WRDC-TM-90-320-FIMM

PRIMER AND ANALYSIS TO EM-TRANAIR CODE EXECUTION

G. S. MESERVE

AERODYNAMIC METHODS GROUP
AERODYNAMICS & AIRFRAME BRANCH
AEROMECHANICS DIVISION

MAY 1990

Approved for public release; distribution is unlimited.

FLIGHT DYNAMICS LABORATORY
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6553

FOREWORD

This report was prepared by Gregory S. Meserve of the Aerodynamics and Airframe Branch, Aeromechanics Division, Flight Dynamics Laboratory, Wright Research and Development Center, Wright-Patterson AFB, Ohio, under Project 2404, Aeromechanics Technology, Task 240410, Aerodynamics/Airframe Technology, Unit 240410A1, Aerodynamic Design and Analysis Methods.

A special thanks to the members of the Aerodynamics Methods Group for their help and patience towards aiding in this effort. The input from J.E. Bussoletti used to assist in jumping the hurdles is greatly appreciated. Don Kinsey's guidance and for being a caring sounding board were invaluable towards reaching successful goals. And finally a hats off to Deborah Rice and Cheryl Brown for helping to create this document.

The work reported herein was performed during the period 29 Nov 89 to 31 May 90.

This work has been reviewed and is approved.

G.S. MESER , Capt, USAF
Aeronautic. Engineer

DR DON KINSEY
Technical Manager
Aerodynamics Methods Group

DENNIS SEDLOCK
Chief, Aerodynamics & Airframe Branch
Aeromechanics Division

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

INTRODUCTION

In simplest terms EM-TRANAIR is a computer program for the solution of Maxwell's equations in three dimensions. The ability to execute the code for the Aerodynamic Methods Group marks the beginning of the investigation into low observable technology. EM-TRANAIR was developed by the Boeing Military Airplane Company and delivered to the USAF in May 1987.

"EM-TRANAIR uses a variational principle to characterize Maxwell's field equations and specified boundary conditions as an extremum problem in the calculus of variations. A multiply-preconditioned Krylov space method called GMRES (generalized minimum residual algorithm), with both an extremely powerful sparse solver preconditioner and an exterior Helmholtz solver preconditioner, provides convergence to machine error in ten to twenty iterations.[1]"

Some schemes used to calculate low observable parameters initially determine the point sources on the surface grid generated by the electric or magnetic field. From these point sources the far field parameters are calculated with a numerical analysis. Through a different approach using prescribed mathematical tools, EM-TRANAIR uses sources that are fundamental unknowns for the constant coefficient differential operator governing the far field, rather than the electric or magnetic field at grid points. The solution is peculiar to the numerical analysis of Maxwell's equations.

This report is intended to relate information concerning manipulation and execution of the EM-TRANAIR code. The process of code installation became as involved as the actual theory and mathematical analysis employed to achieve solution. Hopefully the following content will prove to be a useful tool to troubleshoot the EM-TRANAIR software.

EM-TRANAIR consists of two main programs supported by five libraries. The two main programs are an input processor, V1INP, and a solver/post-processor, V1SOL. The input processor is extremely extensive in length which makes it difficult to troubleshoot for errors. It's quite time intensive when it comes to gaining familiarity.

The library, input and solver files are first saved as program listings and binary files using UPDATE. "UPDATE is a line oriented test editor for maintaining programs in the form of source code, as well as other types of text data. UPDATE creates and modifies program libraries and produces output that can be used as input to other programs, particularly compilers and assemblers.[2]"

The library files are needed so that pieces may be pulled from them when the source decks for the input processor and the solver are created using SEGLDR. The reason for this methodology is to economize the available central memory during code execution. Use of SEGLDR results in the creation of a nonsegmented executable program. One for the input processor called EMINP and one for the solver called EMSOL will be created. Nonsegmented programs are those that have all of their code continually memory resident. "SEGLDR is an automatic loader for code produced by language processors.[3]"

The code delivered by Boeing was compiled using CFT FORTRAN compiler on a CRAY computer with 8 million words of available central memory under the COS operating system. These three characteristics (chosen compiler, size of central memory and chosen operating system) are key factors influencing successful code execution.

Execution problems needed to be resolved to utilize EM-TRANAIR and obtain useful numerical RCS calculations. Compiler problems such as the ability to pass argument values between source programs and subroutines are addressed. Hardware limitation such as the size of the central core memory is an important factor in what configurations would result in converged solutions. Manipulations within the FORTRAN codes such as setting initial values for variables and memory allocation for vector arrays are also discussed. Some familiarity with successful numerical RCS calculations resulted in the presentation of a few "rules of thumb." These and other installation and execution problems are presented to help the reader with EM-TRANAIR.

This document is not a dissertation on theory and code development but a helpful primer to EM-TRANAIR installation at WRDC and a guide to trouble-shooting confronted execution and processing problems.

INPUT DATA/SOURCE DECK

GFLIB, EMLIB, GPLIB, TRLIB and EXSOL are the libraries that support the input processor and the solver. GFLIB calculates the Green's function. EMLIB contains the EM (electromagnetic) subroutines. GPLIB contains the general purpose subroutines. TRLIB contains supplemental subroutines. EXSOL contains the exterior Helmholtz solver. The subroutines within the libraries, the input processor and the solver are contained in Appendix A.

Appendix B shows a listing generated when the input processor V1INP was initially compiled using CFT77. Because of the difference between the CFT and CFT77 compilers a number of bugs were created and needed to be resolved for proper compilation. Equivalence and common statements left a number of errors as can be seen in the listing in Appendix B. The CFT77 compiler didn't like the way variables were being passed from the source code back and forth to the subroutines. The call and subroutine statements involving these passed variables were extended to fully cover all the arguments. In this way the equivalence and common statements could be deleted from the code in certain places. The best and recommended way to find the changes would be to compare the original deck with the executable deck. It may be a time consuming task, but the decks are just too extensive to create listings and the number of changes were numerous. The listing such as in Appendix B will provide the information so one can zero in on an area of interest. Remember, the listing in Appendix B can be generated for any code with ON=H in the CFT77 job control statement.

Another problem encountered with compiling on CFT77 was the dimensioning of arrays. The pointers for the arrays weren't being calculated as integers but real numbers. These problems were fixed by insuring arithemic operations dealing with array pointers resulted in integers.

The final problem worth mentioning dealt will boolean algebra operations. Only the CFT77 can do boolean manipulations. Makes you wonder how the Boeing folks got around this one using CFT? The bottomline is that an executable deck containing boolean algebra requests can only be compiled using CFT77 to be successful at WRDC.

UNICOS operating system requires the use of the CFT77 compiler. Half the battle is won switching from COS to UNICOS because hopefully most of the bugs have been resolved. The code has been made CFT77 compile successful although being originally compiled on CFT.

Just a little hint on troubleshooting, the $DEBUG and dump files are a great aid in getting traceback maps. These maps help in following paths to that place in the code that tripped the execution error. Using COS or UNICOS, various flags are turned on in the job control language statements to establish symbol files and maps for $DEBUG.

Happy hunting.

V1INP INPUT PROCESSOR

The input processor is the heart of EM-TRANAIR. The code contains approximately 27,000 lines making path following to debug problems quite a task. The nonsegmented structure helps in your searches for execution problems because the numerous subroutines plus the libraries subroutines act as blaze markings along the pathway. The imbedded error flags and output messages are also another big plus.

The processor reads in the case test configuration and does extensive presolving tests and comparisons to avoid glitches when it's time to use the solver. "The V1INP program reads a description of a configuration and some incidence conditions describing a plane wave and computes coefficients of a discrete operator on a Cartesian grid which approximates Maxwell's equations and appropriate configuration boundary conditions.

The functional operations of the input processor are:

- o Process User Input Data
- o Assemble Configuration and Test for Validity
- o Define Coupling with Cartesian Grid
- o Define Locations of Unknowns
- o Compute Discrete Operators
- o Communicate Problem Definition to Solver, V1SOL [4]."

The first problem and probably the most recurring is central memory management. Because the size available was limited to under 2 million words of memory the choice of a test case was highly restrictive. Various cases run by Boeing are discussed in reference [1] and one of these would be a good example to compare against during code installation execution runs. It became fairly apparent that only the sphere would become the likely candidate.

The key driver is choosing a size for the parameter NDUMSS. The default values or really the values set in the delivered code established for NDUMSS and other memory management parameters are listed in Table 1. NDUMSS sets the amount of scratch storage needed for the input processor to do its compilations and manipulations. Using NDUMSS set to 1.1 million words really limits the amount of remaining central memory for the other parameters when less than 2 million words total is available. With this in mind and discussions with Boeing, NDUMSS was set to 800,000 words and the sphere would be the only test case that might fit.

The sphere configuration was still to large to fit on the CRAY using COS. The solution was to pull out every other point of the surface grid. This resulted in an executable test configuration. Using COS and less than 2 million words central memory resulted in a really simple geometry for the initial code installation. Memory management is highly dependent upon input configuration.

Variable	Value	Description
MXNETT	225	Maximum number of networks in configuration
MAXPTS	7500	Maximum number of mesh points.
NGRD	140481	Maximum number of grid points.
MAXSRC	14048	Maximum number of sources, any one component
NDUMSS	1.1M	Total scratch storage
MAXSRCH	100	Maximum number of search directions (GMRES) any one component
MAXMAT	100	Maximum number of different materials
MXLPAN	150	Maximum number of panels in a grid box
MXNTPN	2000	Maximum number of panels in one network
MAXNB	14048	Maximum number of B points (not used)
MXIABT	3200	Maximum of pairwise abutments
MXNABT	1000	Maximum number of abutments
MXEIAB	20	Maximum number of edges in an abutment
MXFDSG	1200	Maximum number of fundamental segments
MXEMPT	6500	Maximum number of edge mesh points
MXEDMP	400	Maximum number of edge mesh points per network
MXNAI	350	Maximum number of abutment intersections
MXNPEC	100	Maximum number of pts in an eq. class
MNODMX	100	Maximum number of nodes at abutment intersection
NSEGMX	60	Maximum number of segments at abutment intersection
MAXKMS	19500	Maximum number of micro-edge abutment descriptions

TABLE 1: EM-TRANAIR Parameter Declarations

Another area that dictated concern dealt with the initialization of code parameters and variables. For some unknown reason the original decks for the input processor and the solver wouldn't execute properly with the initial value sets for the variables. For the processor and solver, source program AAINPUT and AA3DS respectively, would call up a subroutine containing the initial value sets. Appendix C contains the listing of the initial value set for the input processor.

To get around this execution problem with the passing of the initial values the contents of the initial values subroutine was imbedded into the source program. This way the transfer with the call and return was eliminated. If the BLOCK DATA statement was left in the source program the execution would still stop. So the BLOCK DATA statement was also eliminated.

When the "new" deck with the imbedded initialization values is run a caution is returned during the CFT77 compile operation. The caution is to reveal that with the initialized values imbedded in the source program there's no guarantee that the initialized value will be kept throughout the program. In the original deck the initialize subroutine is only called once so its pretty safe to say no problems are encountered due to initialization imbedded in the program source deck. The imbedded initialization to the input processor V1INP program source deck AAINPUT is also shown in Appendix C.

As stated previously, central core memory management was a recurring problem. After getting over the variable initialization problem the input processor still wouldn't execute too far down into the deck. A bit of time was consumed troubleshooting and the best means of solution was to track the values of various variables using write statements to print out values on FT06. It was discovered that the surface grid network for the conducting sphere had twice as many points as there was space provided for in the vector arrays. There weren't any imbedded helpful error messages in V1INP to point at a fix to the problem. Since the size of central memory was set, the number of points on the surface grid had to be decreased. Every other point was eliminated on the surface grid input data. The original surface network grid and final network grid for the conducting sphere configuration are shown in Appendix D. Using the UNICOS operating system instead of COS because of more available central memory should alleviate any problems with surface grid sizes.

One oversight to avoid when trying to follow tracebacks concerns calls to subroutines that won't be found in the listings of Appendix A. It's not that they don't exist but they are CRAY libraries. A couple of discovered examples where CRAY subroutines are called upon are as follows.

The library deck GPLIB which contains the general purpose subroutines also has a call statement for WOPEN. When the subroutine listings for the libraries in Appendix A are reviewed it will be discovered that there is no listing for WOPEN. This is an internal CRAY library subroutine that opens a word-addressable, random access dataset.

The same can be discovered for the routine SCOPY found in the input processor deck, subroutine ABTIDN. SCOPY, again an internal CRAY library subroutine, is used to copy a real or complex vector into another.

Tracebacks greatly aid in path following troubleshooting. If the path appears to call on an unlisted subroutine not found in Appendix A, it's probably a CRAY internal library subroutine.

The various problems encountered by the author to get the input processor V1INP to execute have been discussed. If any hints or insights were revealed to the next individual working with EM-TRANAIR the reward was well worth the effort. Use this information to work the changes made to the original decks if different test configurations result in unexpected execution errors and unrealistic computational results.

V1SOL SOLVER

With the input processor being the heart of EM-TRANAIR, the solver, V1SOL is the life support. Extensive preprocessing with the input processor establishes an input configuration that predictably runs through the solver error free. By comparison the input processor contains approximately 27,000 lines of code while the solver contains approximately 3,000 lines of code.

The solver, V1SOL, "relies heavily on two special purpose libraries to provide the discrete Green's function and the exterior Helmholtz Solver (GFLIB and EXLIB, respectively).[4]"

"The V1SOL program iteratively solves the set of discrete equations which approximate Maxwell's equations and describes the scattering of electromagnetic energy from some material object. The solution process can be divided into the tasks of:

- o Initialization
- o Data Entry
- o Overhead Operations
- o The Iterative Solution Process [4]"

As with the input processor, the problem of initialization of parameters was also encountered with the solver. Two subroutines, INIT and INICOM (shown in Appendix E), in the original V1SOL deck were used to pass parameter initial values into the source program AA3DS. The source program AA3DS would call on INIT which would call on INICOM. As expected, the original deck wouldn't execute. To resolve the problem, subroutine INICOM was imbedded into subroutine INIT which in turn was imbedded into the source program AA3DS.

The results are shown in the second listing of Appendix E. This listing is the revised section of the source program AA3DS that contains the imbedded parameter initialization.

The only other execution error encountered with the V1SOL was due to a couple of vector arrays being erased from allocated central memory. The vector array was then needed for manipulation further into the code but was no longer available. This problem occurs in subroutines CLUSS and CLUSM which are found in V1SOL and called from the source program AA3DS. The subroutines are listed in Appendix F along with a "dummy" subroutine called IEQUAL.

Vector arrays CB and KCB are the ones that are inadvertently erased. In subroutine CLUSS, vector array KCB is equated with vector array NC. When this is done vector array KCB becomes unlisted on central memory because that allocation of memory space has become known as NC. This is where the IEQUAL "dummy" subroutine comes in. By passing the vector arrays CB and KCB as arguments their memory allocation remains intact. If this isn't done the computer can't find the arrays because essentially they don't exist. But

with IEQUAL, vector arrays CB and KCB still retain their values and memory allocations.

Why are both vector arrays, CB and KCB, passed as arguments to subroutine IEQUAL? Depending on subroutine CLUSS and CLUSM, there are statements within these subroutines that equate CB and KCB to other vector arrays. Note Appendix F is what code exists in the revised V1SOL deck. IEQUAL didn't appear in the original V1SOL deck.

Installing the solver was a small hurdle compared to installing the input processor. Hopefully, the insights given to troubleshooting these execution errors will aid in any future troubleshooting. These solutions may be a place to start an investigation for future solver software problems if encountered.

CONFIGURATION EXECUTION PROCESSING

Before the SSD on the CRAY was committed as an external resource for UNICOS a number of configurations where submitted for execution. This section deals with a few "rules-of-thumb" that were discovered with this opportunity to test the code while the COS operating system was still available. These rules-of-thumb are still applicable with the UNICOS operating system because they are peculiar to EM-TRANAIR.

When execution starts in the solver, V1SOL, various parameters are computed and output to FT06. Two of these, delta (for X, Y and Z) and the number of mesh points per wavelength, lambda, are especially important. Delta should be within the range

$$.2 < \text{delta} < 5$$

and the number of mesh points per wavelength, lambda, should be within the range

$$5 < \text{points}/\lambda < 10$$

If you stay within these limits the numerical results are reliable.

The Green's function is calculated within V1SOL. Information concerning the Green's function is output to FT06. Because of some bug in the code a flag stays on that causes error tolerances to be printed even if the calculations are within accurate solution boundaries. Note that if the tolerances output to FT06 are zeros; rest assured the Green's function is accurate.

Beware of thin geometries such as disks and flat plates. The delta aspect ratio in the "thin" direction may lead to problems in getting a converged solution to Maxwell's equations.

A "thin" configuration resulting in a solution is greatly influenced by the chosen frequency used for the incidence radiation. If a low frequency is specified, you won't have many mesh points per wavelength, lambda, in the "thin" direction. If a high frequency is specified you'll be playing the aspect ratio deltas against the calculated discrete wave number. A possible solution to "thin" geometries may be to scale up the geometry ($REFerence length parameter) so the "thin" dimension is no longer a problem.

Finally, be careful with the $BOX definition parameters. Be sure the number of mesh points in the computational grid is of the type

$$(2^a)(3^b)(5^c)+1$$

Where a, b and c are integers and the resulting number is odd.

These "rules-of-thumb" will be helpful in achieving converged solutions. Experience with the code is the best medicine but the rules-of-thumb are a good heads up before diving in.

POST RCS RESULTS

Before switching the SSD over to UNICOS a few weeks were given where 4 million words of central memory were available with COS. This was an excellent opportunity to try some EM-TRANAIR runs before having to change to UNICOS. This was the chance to gain some familiarity with the code and become better assured things were executing properly.

Various configurations were tried and some were successful and others ran into problems as were outlined in the previous section concerned with configuration execution processing. Presented here are some actual post RCS graphics that compare results to those conducted by Boeing and an actual comparison to an experimental run taken at one of ASD/WRDC radar ranges. These preliminary results are encouraging and reveal the acceptable reliability concerning the software code EM-TRANAIR.

The first configuration used was a conducting sphere. This configuration is the same as used by Boeing for code validation and a precise definition is presented in reference [1,4]. Figure 1 was taken from reference [1]. Figure 2 shows the results for the same conditions as Figure 1 but being an execution run on WPAFB's CRAY X-MP/216. A total RCS measurement for vertical polarization incidence at 2 degree interval bistatic stations is calculated. A fifth order polynomial curve fit was used to create the smooth fit through 90 bistatic stations. The agreement is excellent.

The frequency of the incident radiation is the driver that regiments acceptable and non-acceptable configuration runs as far as number of mesh grid points and points per wavelength lambda. Numerous configurations were tried to get a better feel for the driving parameters and the number of failures exceeded the number of successes. The size of an acceptable geometry is determined by the incidence radiation frequency. This relationship is the key to an accurate solution of Maxwell's equations using Green's function.

Initial runs were executed at high incident frequencies from 9 to 16 GHz. The geometries being used just weren't compatible to the desired incidence frequencies. Possibly, the problem was being to close and in some cases being involved within the resonance region. Looking over the evidence mostly the Rayleigh region appears to lead to code execution success.

The bottom line for the preliminary lessons learned revealed that staying within the Rayleigh region brought about accurate solutions and this is the norm for the configurations used by Boeing for validation. So more configurations were executed to try to mimic Boeing's results.

Moving up in incidence frequency, Figure 3, taken from reference [1] is Boeing's run with a conducting sphere at a medium frequency as determined by the diameter of the sphere. The execution results on the CRAY X-MP/216 for the same configuration used for Figure 3 are presented in Figure 4. A total RCS measurement for vertical polarization incidence at 2 degree interval bistatic stations was calculated. A fifth order polynomial curve fit was used to create the smooth fit through 90 bistatic stations. The agreement

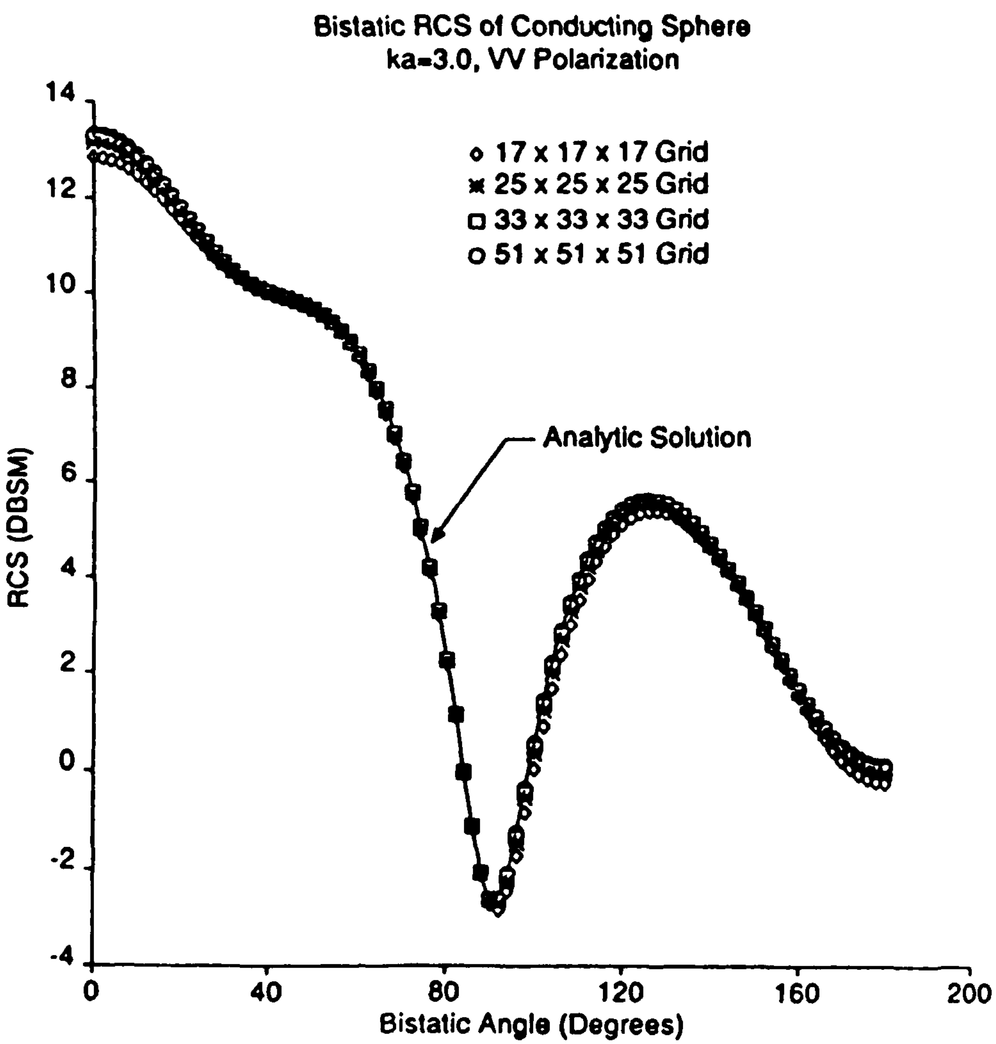

FIGURE 1: Vertical Polarization Bistatic RCS of Low Frequency Sphere (Boeing)

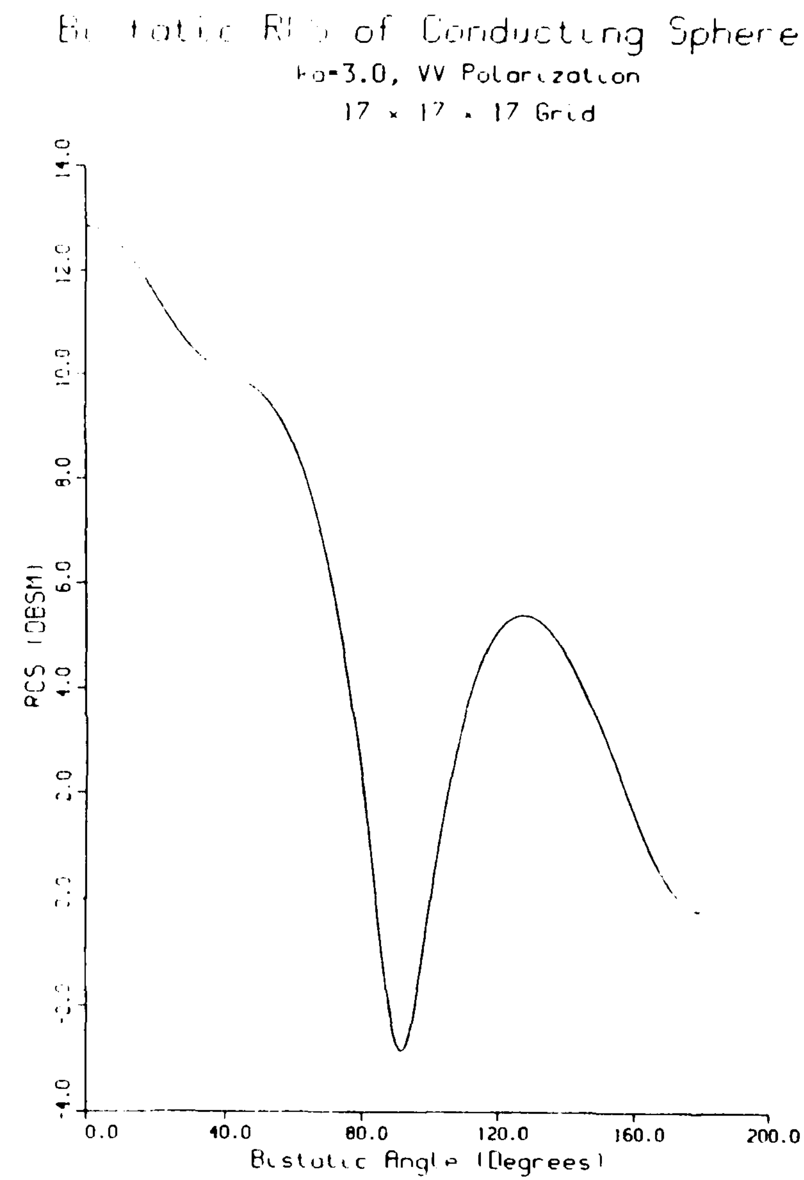

FIGURE 2: Vertical Polarization Bistatic RCS of Low Frequency Sphere (CRAY X-MP/216)

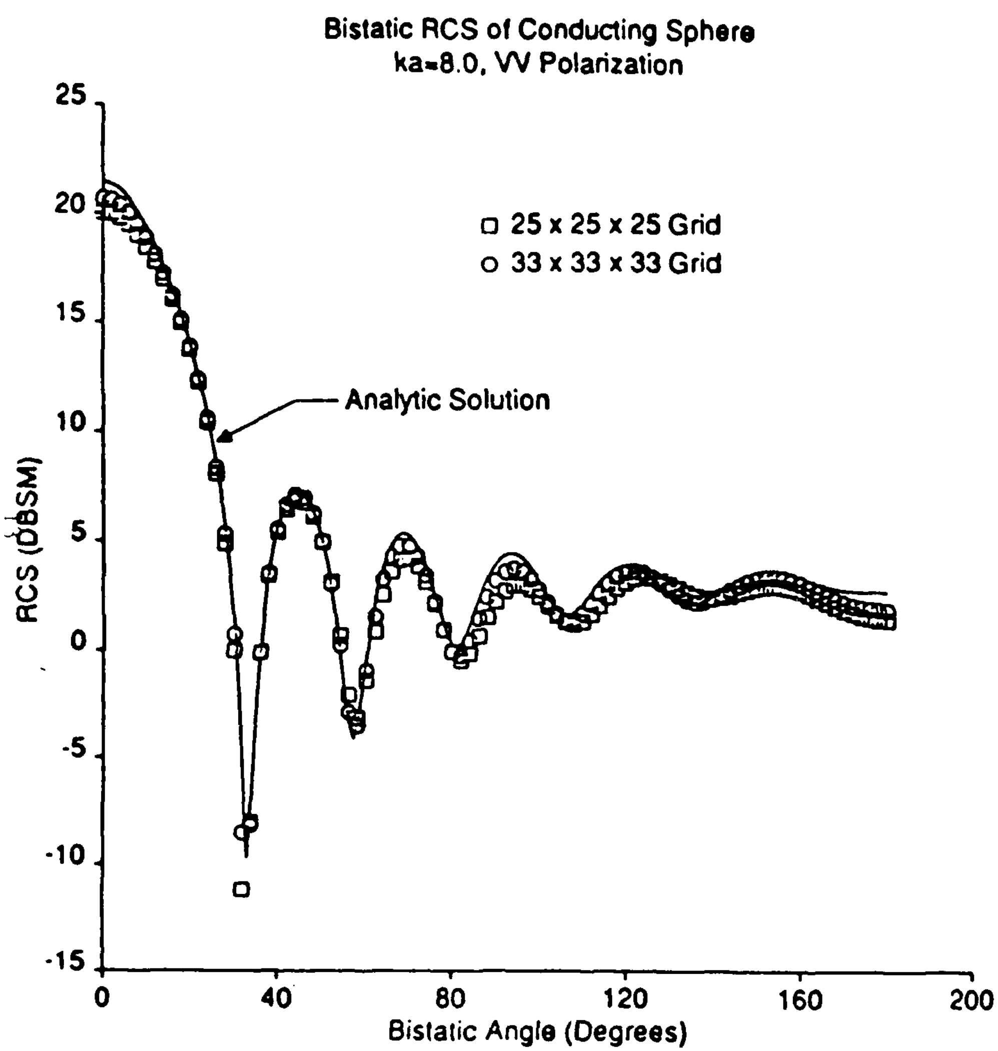

FIGURE 3: Vertical Polarization Bistatic RCS of Medium Frequency Sphere (Boeing)

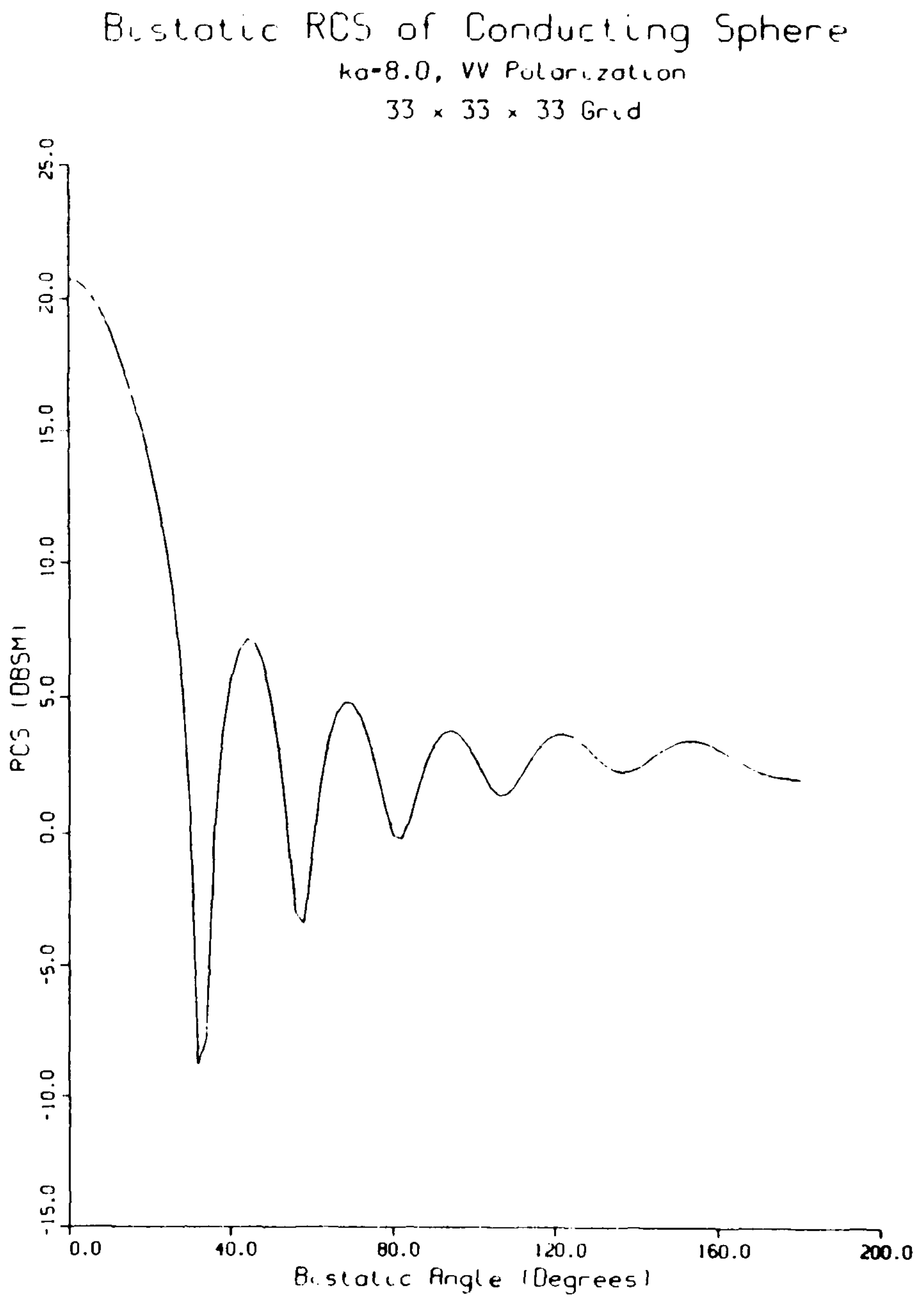

FIGURE 4: Vertical Polarization Bistatic RCS of Medium Frequency Sphere (CRAY X-MP/216)

looks good with the possible exceptions of the deep troughs. More bistatic calculations, say at every degree interval, would produce a better curve.

Having moved up in incidence frequency, a look was taken to see what type of agreement was found at the same incidence frequency but with horizontal polarization of the incidence electric field. Figure 5 was taken from reference [1]. Figure 6 shows the results for the same conditions as Figure 5 but being an execution run on the CRAY X-MP/216. A total RCS measurement for horizontal polarization incidence at 2 degree interval bistatic stations was calculated. A fifth order polynomial curve fit was used to create the smooth fit through 90 bistatic stations. The agreement is excellent.

Finishing up the conducting sphere configuration successful results, is a look at monostatic RCS measurements. Figure 7 shows the monostatic RCS measurement for a 6-inch diameter conducting sphere being radiated by a vertical polarized electric field at 10 GHz. This actual experimental run is presented to show how the total RCS magnitude is a constant over the surface of the conducting sphere. A good numerical result using EM-TRANAIR to compare to the experimental measurements in Figure 7 was not achieved. A good guess as to the inaccuracies in the Green's function solution of Maxwell's equations may be due to the high frequency incidence radiation and the small diameter geometry. A converged solution couldn't be obtained. The sphere configuration used as Boeing's validation case was used to obtain a numerical monostatic RCS solution (the low frequency case with ka=3.0). The important comparison between the numerical results and the experimental results in Figure 7 is the constant magnitude of RCS over the sphere surface. Figure 8 shows the total RCS measurement for vertical polarization incidence at 2 degree interval monostatic stations. A fifth order polynomial curve fit was used to create a smooth fit through 45 monostatic stations. Figure 9 shows the total RCS measurement for horizontal polarization incidence at 2 degree interval monostatic stations. Once again, a fifth order polynomial curve fit was used to create a smooth fit through 45 monostatic stations. The monostatic RCS numerical results with EM-TRANAIR definitely possess a constant RCS magnitude over the conducting sphere surface for both a horizontal and vertical polarized incidence electric field.

The last configuration and numerical RCS result to be discussed was an attempt to get RCS information on an 8-inch diameter, .25-inch thick, conducting circular disk. The reason this configuration was chosen is due to the experimental results made available from one of WPAFB radar ranges. These experimental results using an incidence radiation (vertical polarization) at 6 GHz are shown in Figure 10. The EM-TRANAIR numerical results are shown in Figure 11 for the conducting circular disk. A total RCS measurement for vertical polarization incidence at 2 degree interval monostatic stations are calculated. A fifth order polynominal curve fit was used to create a smooth fit through 45 monostatic stations. The Radar Range (Figure 10) angle in degrees from 90 to 180 degrees should be compared to the X-MP/216 (Figure 11) angle in degrees from 0 to 90 degrees. This comparable range runs from a face on monostatic view of the disk revolving to an edge on monostatic view. Magnitude in DB from Figure 10 is the same as Total RCS in Figure 11. The comparison between the experimental and

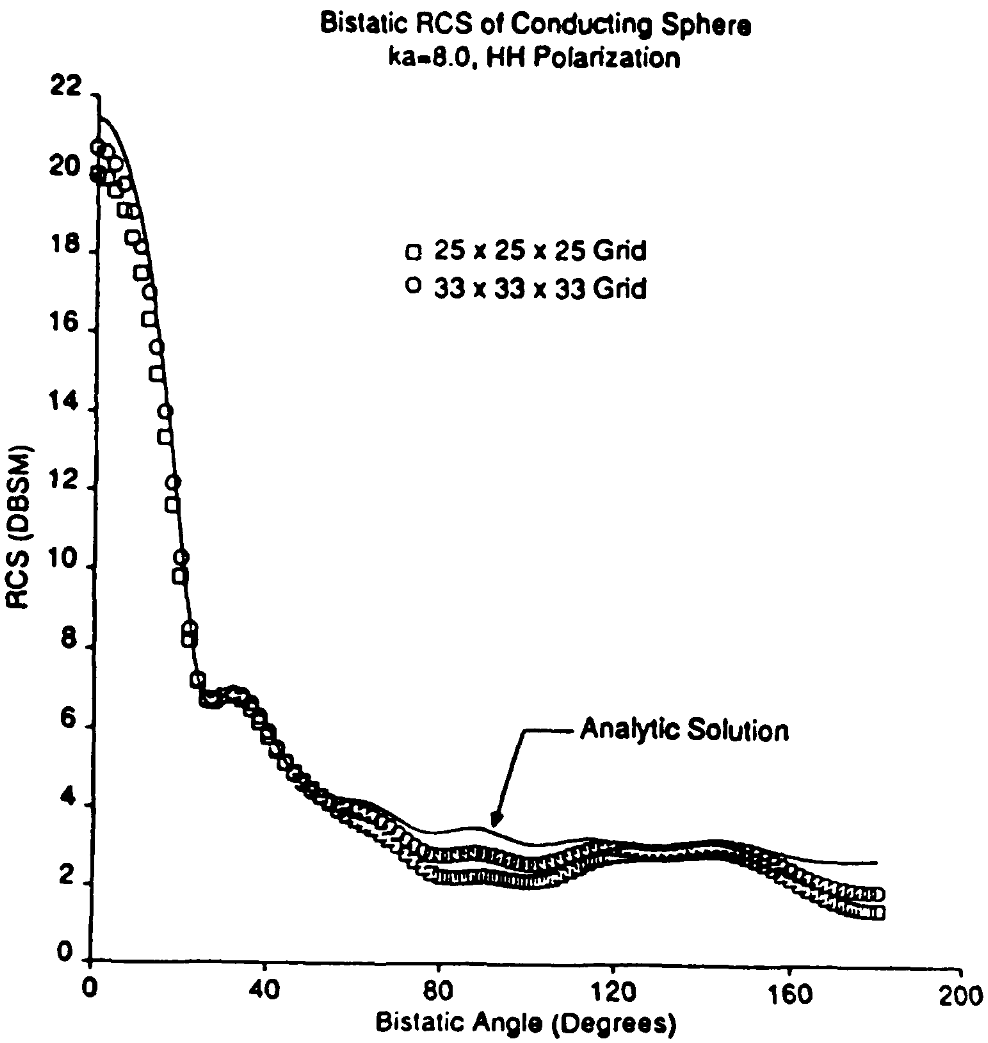

FIGURE 5: Horizontal Polarization Bistatic RCS of Medium Frequency Sphere (Boeing)

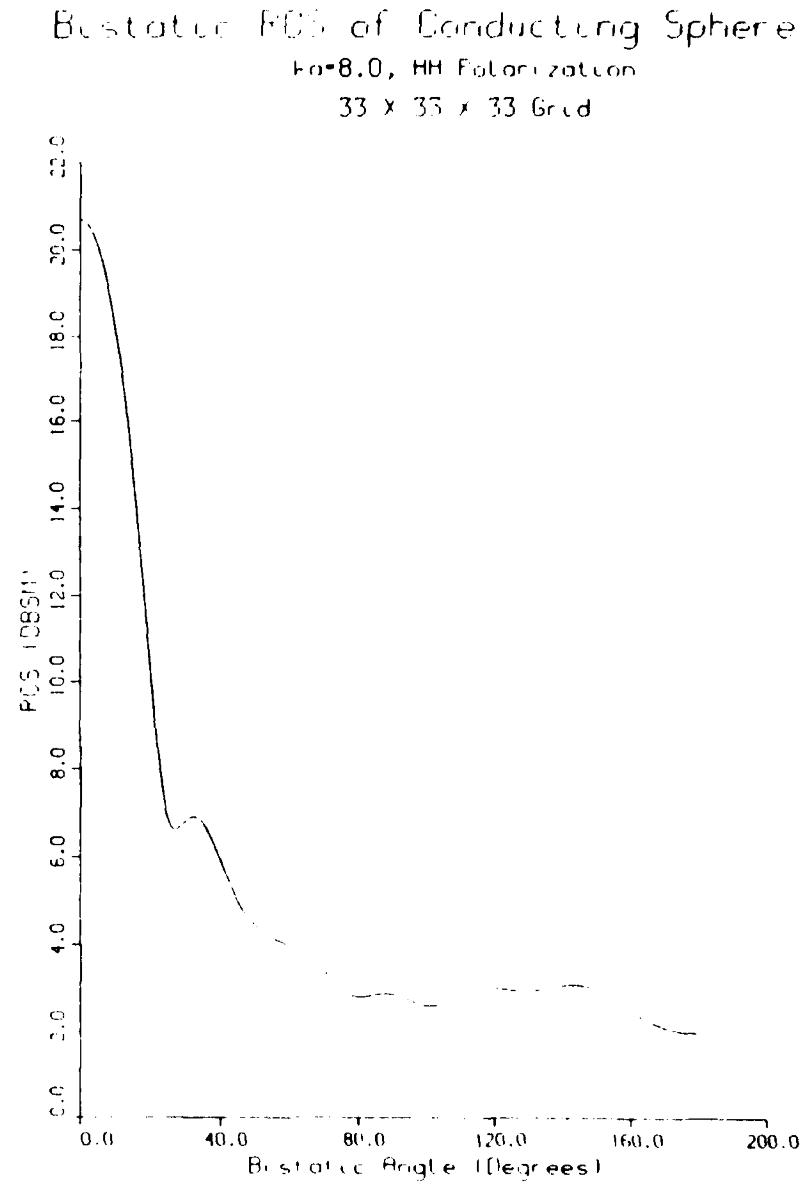

FIGURE 6: Horizontal Polarization Bistatic RCS of Medium Frequency Sphere (CRAY X-MP/216)

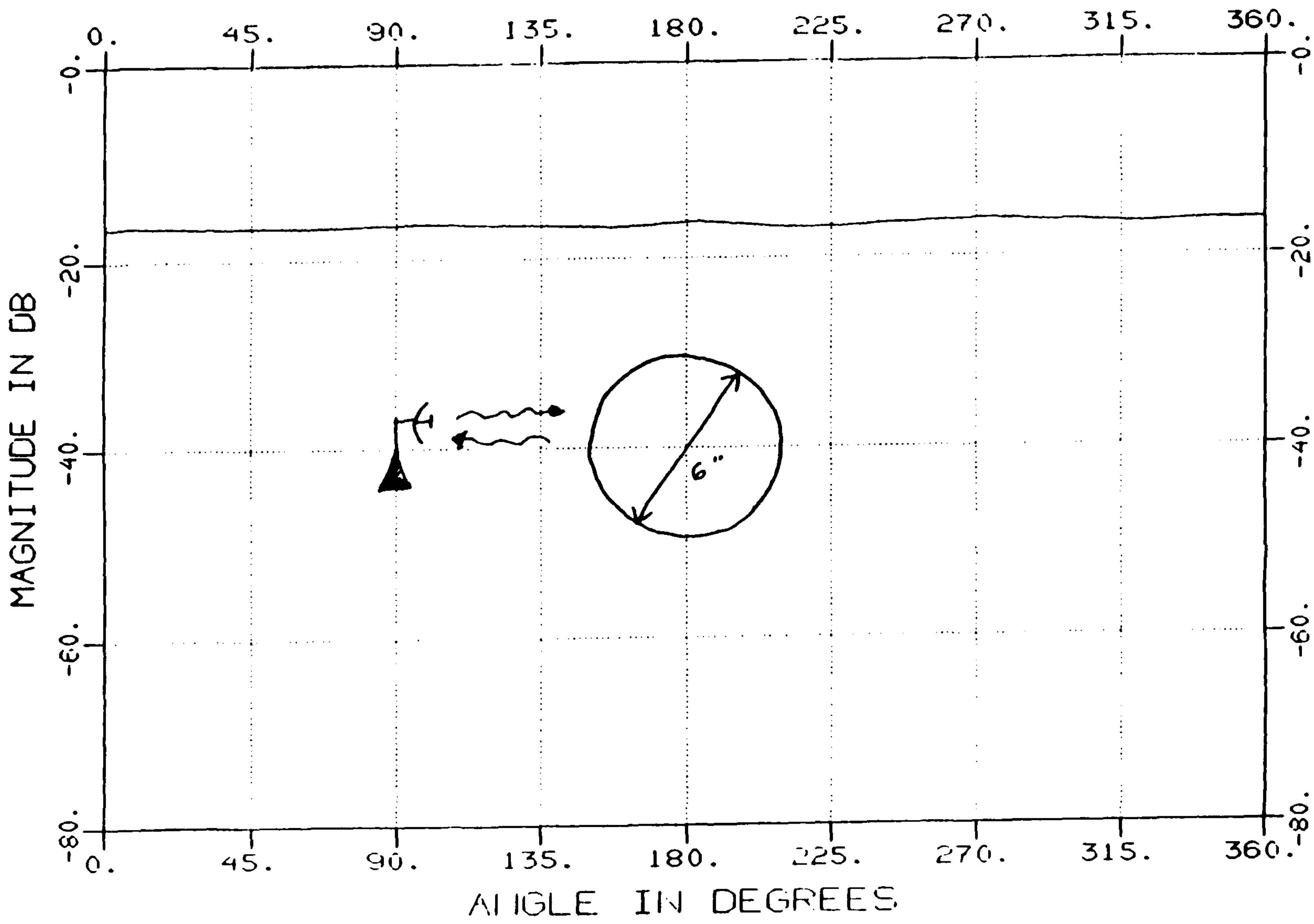

FIGURE 7: ASD/WRDC Radar Range 6-inch Diameter Conducting Sphere Test Case
Vertical Polarization Monostatic RCS at 10 GHz

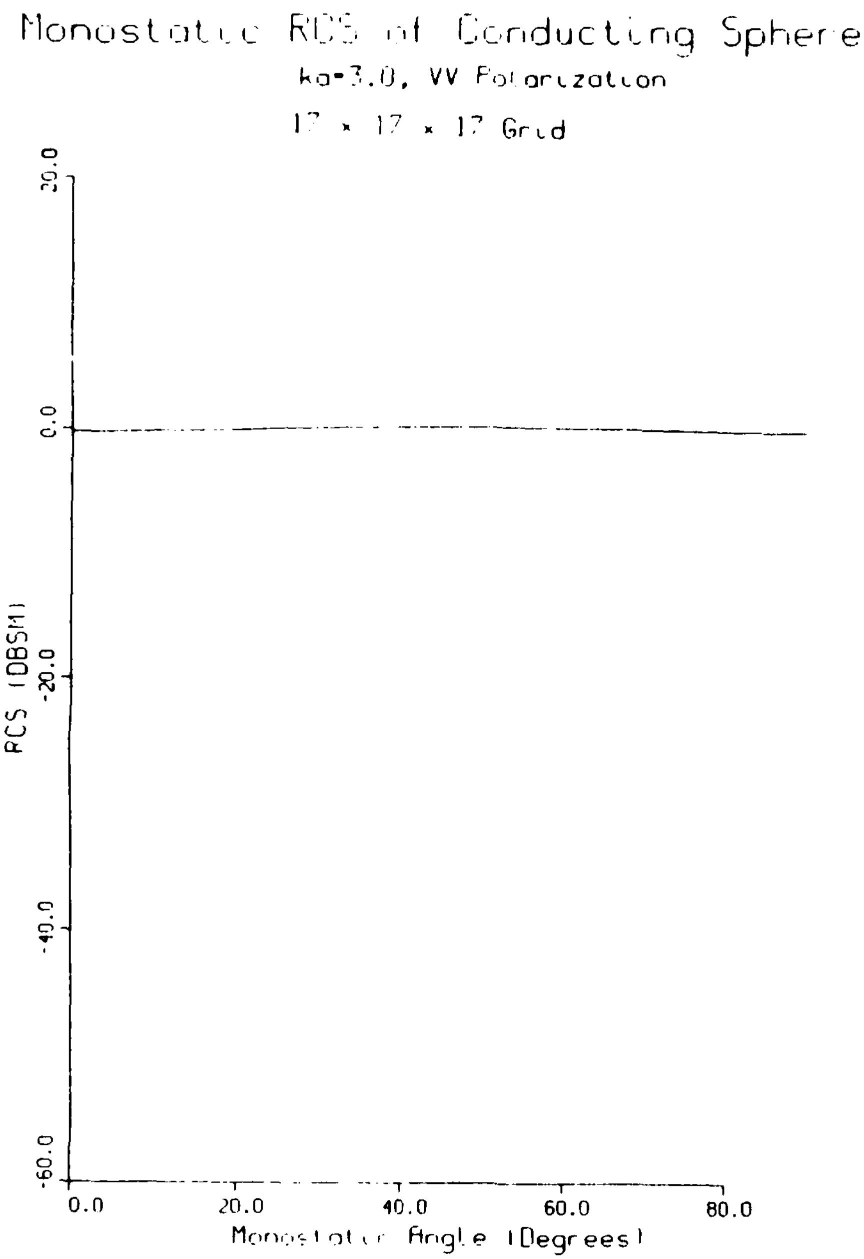

FIGURE 8: Vertical Polarization Monostatic RCS of Low Frequency Sphere (CRAY X-MP/216)

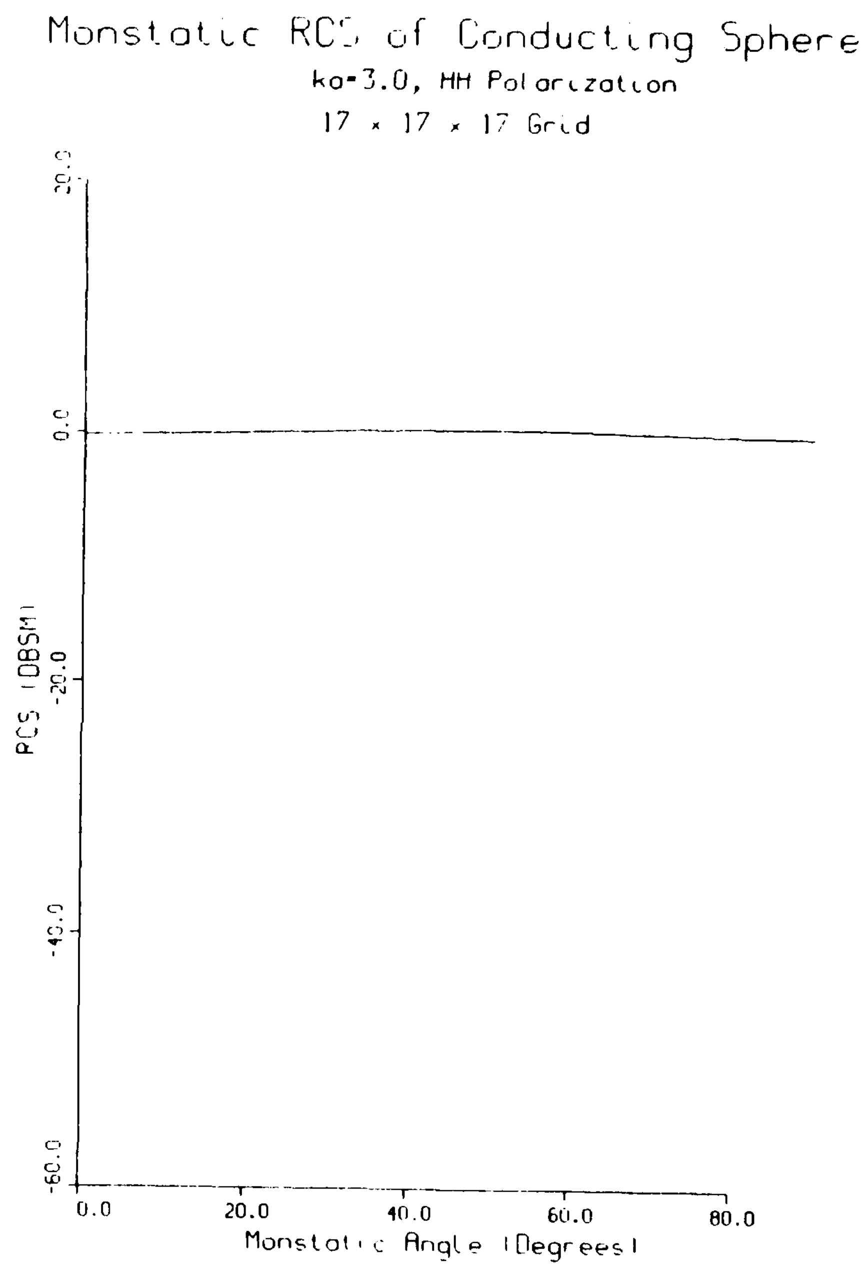

FIGURE 9: Horizontal Polarization Monostatic RCS of Low Frequency Sphere (CRAY X-MP/216)

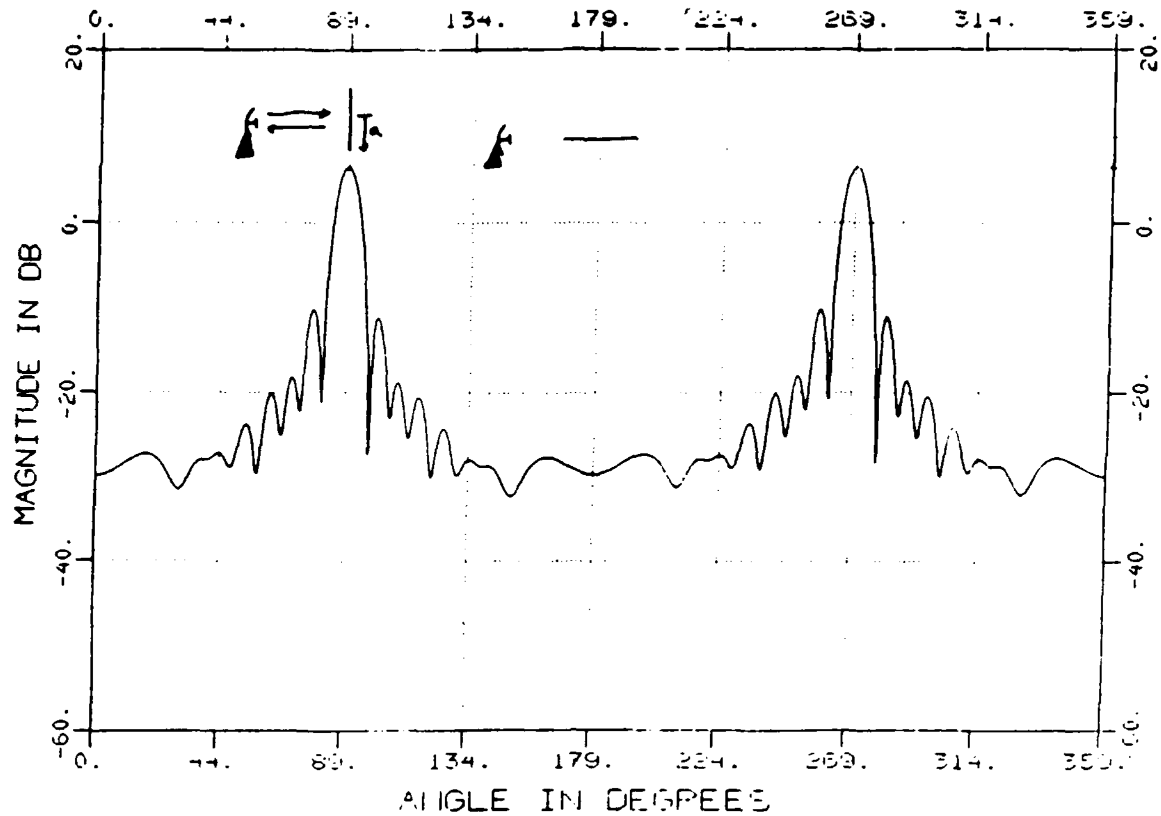

FIGURE 10: ASD/WRDC Radar Range 8-inch Diameter .25-inch Thick Circular Disk Test Case Vertical Polarization Monostatic RCS at 6 GHz

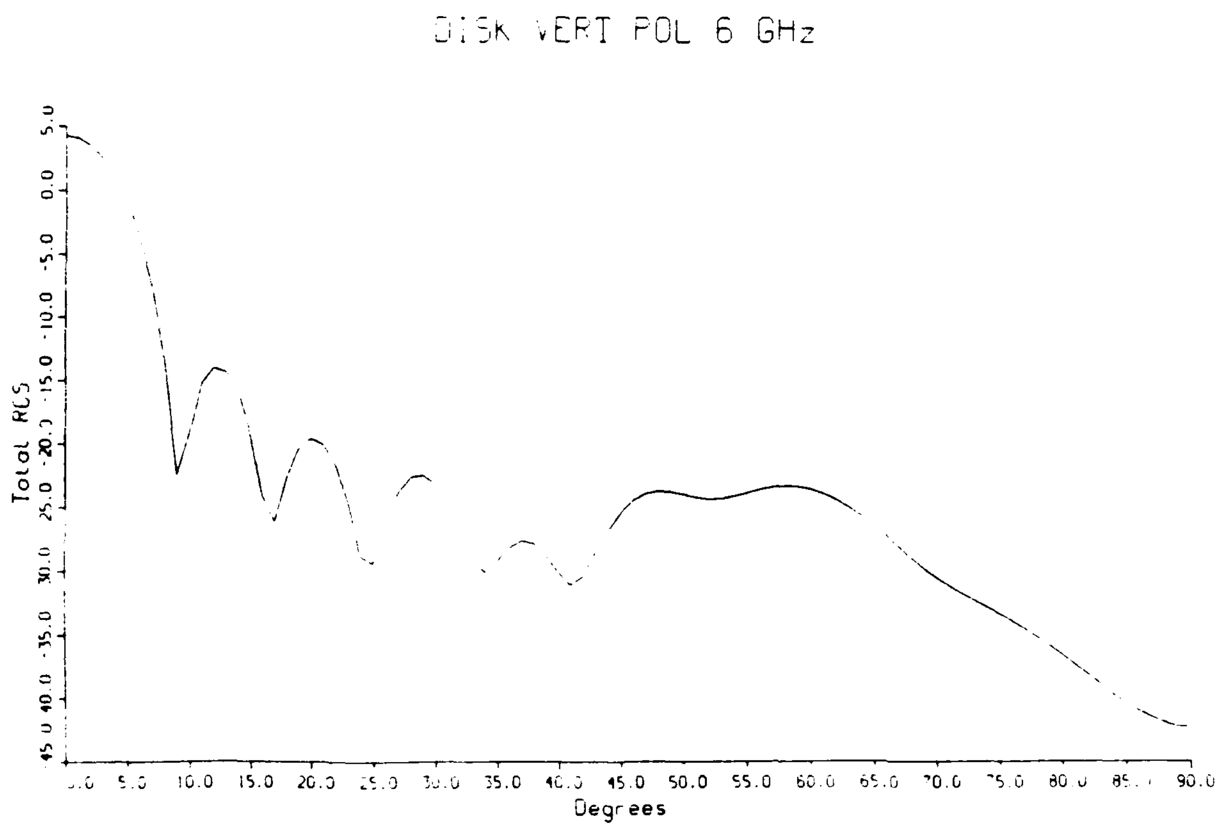

FIGURE 11: 8-inch Diameter .25-inch Thick Circular Disk Vertical Polarization Monostatic RCS at 6 GHz (CRAY X-MP/216)

numerical results reveals a pretty good correlation between the two when defining the troughs and the nodes. The magnitude of the RCS face on (full side of disk) is in good agreement, but as the measurements near the "thin" edge the similarity falls off. Possibly this is a region that doesn't contain enough mesh points per wavelength, lambda, and/or a good enough aspect ratio to get a good converged solution. In reference [5], user's of EM-TRANAIR at the Naval Weapon Center at China Lake had problems with the numerical solutions on closed cylinders.

"[They]...discovered that solutions for vertically polarized scattering from conducting cylinders differed significantly from experimental measurements and from solutions obtained by other computational techniques." The conducting circular disk, being a closed cylinder may be incurring this inherent problem. More field testing is encouraged.

CONCLUSION

EM-TRANAIR is a computer program for the solution of Maxwell's equations in three dimensions. This primer on EM-TRANAIR installation at WRDC and guide to confronted execution problems will aid in understanding the internals of EM-TRANAIR. Successful pathways for accurate converged numerical solutions are possible.

The architecture of EM-TRANAIR was briefly presented in the discussion of various installation problems encountered when moving from a CFT compiler to a CFT77 compiler. Compilation problems and solutions dealt with passing arguments, dimensioning arrays and boolean algebra manipulations.

Execution problems with the input processor, V1INP and the input solver, V1SOL were presented. Troubleshooting methodologies and working solutions were revealed. The actual problems needing to be surmounted for successful program execution are the examples contained in this document.

Different input configurations were used to establish "rule-of-thumb" parameters to achieve accurate numerical solutions. Parameters such as incidence frequency, number of mesh points and aspect ratios were discussed.

Finally, the area to prove being the most successful in the post-processing examination was the Rayleigh region.

The numerical RCS solutions presented are proof positive that the EM-TRANAIR code is executable and providing good computational RCS results.

REFERENCES

[1] J.E. Bussoletti, F.T. Johnson, K.W. Sidwell, B.L. Everson, D.P. Young, R.H. Burkhart and S.S. Sament, "EM-TRANAIR: A Computer Program for the Solution of Maxwell's Equations in Three Dimensions: Volume I, Theory Manual" Technical Report AFWAL-TR-87-3082, (1987).

[2] CRAY Research Inc., CRAY X-MP and CRAY 1 Computer Systems, "UPDATE Reference Manual (SR-0013)," 1986.

[3] CRAY Research Inc., CRAY X-MP and CRAY 1 Computer Systems, "Segment Leader (SEGLDR) Reference Manual (SR-0066)," 1988.

[4] J.E. Bussoletti, F.T. Johnson, K.W Sidewell, B.L. Everson, D.P. Young, R.H. Burkhart and S.S. Sament, "EM-TRANAIR: A Computer Program for the Solution of Maxwell's Equations in Three Dimensions: Volume II, User's Manual" Technical Report AFWAL-TR-87-3082, (1987).

[5] Boeing Advanced Systems Computational Methods, "EM-TranAir Software BUG Report," December 1989.

APPENDIX A

SUBROUTINE LISTINGS FOR LIBRARIES, INPUT PROCESSOR AND SOLVER

This appendix is intended to guide the user through the various pathways as the result of CALL SUBROUTINE statements. The listings were generated with UPDATE and the CD output option under COS.

*****APPENDIX A - V1INP SUBROUTINE DECKS*****************************

AAINPUT	INIT	FLDFIL	ERRMSG	NETMOD	OPDEF
PIW4	UNPIW4	BOPDEF	EDGPNT	FBFND	
FESBLE	FLDDEF	LPCUT	LTCUT	PANDQC	PANDEF
PANPNT	PFBLST	PIFBCH	TADJST	TCOORD	
TIFBCH	TRIPNT	STRTAU	ABTCHK	CPETP	EDGABT
DETEQV	PROEQV	CHGGRD	PRONEQ	PEADET	
NETWRKS	MSHIND	EMARK	FBXLST	PADJST	PANGHB
EPTAU	ABTIDN	ABTDNC	DNCHEK	MPNORM	
D2LINE	DOMVEC	ABTAIP	ABTDZO	DZBCHK	ABTECD
ABTFSD	ABTDAB	ABTSYM	ABTDAI	ABTDUE	
ABTAIO	ABTABO	FSGCMP	ADDIN2	ABTEMP	ABTEQC
MPTEQC	ABTMSG	ABTEND	CMPIED	EDGIND	
EPOINT	NRMESH	ABTELS	IDNGEO	ABTPOS	EDGMPI
NREDGE	INTNRM	EDGTAU	OUTPKV	BMARK	
PERPNT	FILOPN	KEYINV	OPGEN	PEAFOR	PROMIN
LSOLVE	MINCUT	GRDGEN	MSPNT1	CAMBER	
CIRC	ELLPT	GADNET	TRINP	INBC	INBC1
INBC2	MESHP	QUADNT	TRWAKE	PEAINP	
MSHDNS	INECHO	INSUM	IJKLAT	LATIJK	GRDDEF
ASEGL	ABTSEG	APNTL	ABTPT1	ABTPT2	
CHCOLP	BCONCL	SPBSC	BCOPT	BLCAL	BLCCAL
CMNGRD	CONTRL	CPABT	CPNOR	DASPL	
XDASPL	XDASIN	EDGSPL	GEOMC	REFLOC	SINCD
SING	SNGDEL	SUBPQR	SURPRO	TCNTRL	
TSING	ZCADJ	ENRCHG	GTALAM	NRPTHP	INSIDE
NRPTED	OPSTNC	DEFOPG	OPSET	DEFDPT	
APTDEL	DEFFPT	TFILL	ADJWK	DREGD	VOPCAL
MAPINV	EQUIVC	LSCHOP	EIFBCH	CIFBCH	
TRICAL	ECAL	FCAL	HCAL	SOPCAL	PDRVR
GLOCAT	VELCAL	TBTOP	PEQCL	PINSID	
PNGHBR	INNGBR	EXNGBR	REQC	TREG	PTINT
ABTORD	VMOMC	EMNCUT	CMPCTL	OPPLOT	
DEFPPT	SMOMC	LINTRI	INDCHK	TRICPI	FPNPLT
SELUL	GNRML	FLDCHK	TRINRM	PAREA	
TESTP	OPCHK	MATCHK	FAOPRT	AGNRML	CDRVR
CNTD	VCALF	VCALA	IBSET	DEFCPT	
LINLS	CDRVR1	IBSET1	DFCPT1	VCALAA	LUMPC
LUMPL	LUMPS	LSQSFT	BASEVL	BASISQ	
ABTINT	GPHSCN	GPHPLK	GPLUCK	DERIVE	SITPNT
THNBCN	TKCAL	TREBRN	TRERUT	KID	
FACRDC	FACTOP	TDPEQV	ODPEQV	ANGORD	EDGRDC
PLNEQV	PROJD	SOUT	WRFAOP	WRSTOP	
FREEOP	BDYST	WVCAL	CAMLT2	TZCAL	CORIND
FLDSCT	FSDCOF	HNGHBR	MXMT	OPCOMP	
DMATOP	EMOUT	OPADD	LSTDEF	CONDEF	CJDEF
SRCDEF	RHODEF	GRHDEF	GAMDEF		

*****APPENDIX A - GFLIB SUBROUTINE DECKS*****************************

INICOR	SETCOR	GETCOR	FRECOR	BNQUAD	BVPGEN
BVPG3D	DFNG3D	DFNIJK	DFNSGV	GETGVL	
GQXTND	GRECUR	OUTPKV	GQRG12	G3DABT	GQRG3
TIMING	OMGCON	GRNAXS	COMGEN	LSQSOL	

LSQFBS	RDG3D	VIP	ZERO	DCBHTG	OUTLIN
OUTMAT	OUTVEC	PIW4	UABEND	UNPIW4	
AMCON	SHELLX	SPARSA	SPARSE	SPARSF	SPARSO
SPARSP	SPARSS	POLGEN	ASYG3D		

*****APPENDIX A - EMLIB SUBROUTINE DECKS*****************************

FFZER	SPCSCAL	SPDIV	SPGRAD	DIV	CCONT
CCDISP	CFILLI	FCOPY	CAXPYI	CDOTCI	
CGTHRS	WHENCGT	VHEAD	CASSUM	BUBBLE	QUICK
DIVIDE	CCDSPY				

*****APPENDIX A - GPLIB SUBROUTINE DECKS******************************

CADD	CFILL	CGATHR	CSCATT	CZERO	RSIGN
ICONTR	CCONTUR	CONTUR	CSTPRT	CVECWR	
DSP	DSPPLT	CONJUG	RMATWR	IMATWR	OVECWR
CMATWR	PLTCAL	PPLOT	FSHELL	KEYSRT	
ISHELL	RELIST	SETPT2	SETPT1	CPLOT	ICOPY
IZERO	LSPARI	PARINT	RFILL	RZERO	
LISTN	IADD	NRMARG	CRXIPY	CRXPYI	CXITY
OMATWR	RTENWR	CTENWR	ITENWR	OTENWR	
CROSS	DISTNC	MAG	MNMOD	PIDENT	SHLSRT
UKYSRT	UVECT	VADD-CR	VMUL-CR	IFILL	
CCXIPY	CCXPYI	MODVEC	PJ[illegible]	PRCONT	ANSPEK
CGMRES	GMRES	NGMRES	MTRXTP	UNIONL	
INTRSL	SETPTR	SE[illegible]HR	CMXM	CMXMA	RRXIPY
RRXPYI	RXITY	BSEQR	BSEQW	REVERS	
GWRANA	MSNDX	CQLSF	DCBHTX	LSQSFX	DEFVEC
DCBHT	SGINV	LSQSF	FSINTP	LSQSG	
INCMPR	SCMPKT	SORTAK	TCOF	TRANS	SRCHOL
CPLOT1	MTVCWR	NORCAL	IBSRCH	UVECTM	
BINSCH	INTP1	INTP2	QCOF	NFILL	BISEQF
BISEQW	IIMAX	IIMIN	PCCONT	CHCONT	
ROUND	RREWND	BISEQR	HSMCON	CMUL	HSFFTS
HSFFT1	HSFFT2	HSFFT3	HSFFT4	HSFFT5	
HSFFT6	HSFFT7	HGERR	MSCAL	MSUM	MCOPY
M2COPY	MAXPY	M2AXPY	M2SCAL	MSCOPY	
M2SCPY	M2SBTR	MSBUTR	MBUTR	M2SHFT	MPSUM
M2PSUM	MVADD	MSWAP	H1FFTR	H2FFTR	
MARITH	H1SIN	H1SSIN	H1COS	PERM2P	RCMUL
PERM2	PERMS	PERM2S	M2ZERO	MVADSB	
MZERO	FACTOR	MPERM	M2PERM	HFFTS	HFFTP
H2SIN	H2COS	H2SSIN	MVCMUL	MADSB	
VMCOPY	MOUT	MIOUT	PGOUT	H1SCOS	H2SCOS
SSCOPY	ROPEN	RREAD	RWRIT	RCLOS	
MSHIFT	MSSUM	GETLIN	RTSRCH	TSRCH	RVECWR
IVECWR	PYCONT	ISMX64	SGBSL	MASUM	
MFILL	MVSADD	SGBL	SGBLT	SGBU	SGBUT
INDFND	PYCSRT	CPMAT	IMSHFT	IMFILL	
VNORM2	REINT	SGBFA	SGBCO	SETRA	GETRA
IOFRA	GCOL	FRERA	SETCIO	FNDREC	
REDRA	GACRA	ISCATC	ISCATZ	CNTIP	CNTRP
MAPEQV	LSRUT	TREBRN	TRERUT	KID	

*****APPENDIX A - TRLIB SUBROUTINE DECKS

CCALN	COMPIP	CMPSCL	GCPCAL	GRDIND	ISCAL
MSROTM	PANUNI	SURFIT	XXADJ	KRMOVE	
PROJ	PANMOM	LCHVAR	EDGLS	LPROJ	ROT1
ROT2	SCALE	TRAN	ZMPROJ	ZMERGE	
UNIPAN	POILO	POILY	POIL1	TRI2D	TRI2DS
H1SYM	H2SYM	BGMUL	BGMUL1	GRNDWN	
GBPRO	GBPRY	GBPRYZ	GDB	GRPLAN	GYPLAN
GRADML	GYADML	GDBO	GRNASM	GCOMP	
GREEN	REGGRN	GREEN1	SMGRN	BDYCV1	BXSIZE
BDYCVO	BADD	BCOP	BADDO	BDYCVY	
EXGRN	PERMZ	POISOL	EXPOIO	EXPOIY	EXPOI1
POIFO	POIFY	POIF1	THRMIS		

*****APPENDIX A - EXSOL SUBROUTINE DECKS*****************************

H1SIN	H2SIN	H1FFTR	H2FFTR	PERM	PERM1P
PERM2P	PERMS	MADD	SWTSUM	TRI3H	
TRI2H	TRI2R	PERMC	H1COS	H2COS	CONBOX
CONVLU	CONFFT	GBPRHE	MVADDN	HELMOP	
HELMO	EXHELM	BDYCVH	HCFFTS	HCFFT1	HCFFT2
HCFFT3	HCFFT4	HCFFT5	HCFFT6	HCFFT7	
MSHUFL	M2SHFL	ISIN11	HFIRST	HLAST	GTRANS
MABUTR					

*****APPENDIX A - V1SOL SUBROUTINE DECKS*****************************

AA3DS	ACAL	ALTFSP	AQCAL	BLDNDX	BLDIEL
CGMRES	DDIVG	DERIVE	DGRAD	DJCOLL	
DNSADD	DNSTFL	FLTDNS	FOPEN	GREENF	HSOLVE
INCFAD	INICOM	INIT	JINIT	JSTORE	
JWRITE	OQCAL	OUTPUT	OCVWR	READIN	RDFAOP
SETUP	SOLVER	TCAL	WVCAL	CLUSS	
CLUSM	LCAL	LICAL	CLUSO	CLUSB	BCCAL
OPGET	RADSUB	MDBIAS	ORDER	PROXIM	
RADBC	CORIND	REDSET	ADJAC	OPCOUP	INFOLV
CMPTLW	FNDLV	BLDLW	EQUIVC	MODVEC	
FFRCS	BISRCS	GAPARM	XFMMTX	MONRCS	CJGDJ
MONOUT	EQNRDC	FNDIND	INDFND	INDMRG	
INDSTD	RADELM	RADFND	RADIND	RDCEQN	REORDR
RESTAK	RSINDX	PSIELM	LINORD	MOMEQ	
PLAORD	PLNORD	NESTED	SELPLN	PSHSTK	REFORM
CSINIT	CSINPT	CSFLTR	CSFLSH	CSPREP	
CSSORT	CSSRTB	CSRADX	CSMRGO	CSMRGI	CSADDO
CSADDI	CSCNRC	CSTRNF	CSLU	CSDEC	
DMPRB	FCHRB	INIRB	STORB	SLSTNE	LLSTEQ
SGTHR	CGTHR	CGTHRZ	CSCTR	ROWSCL	
INCFDI	IEQUAL				

APPENDIX B

ERROR MESSAGES GENERATED BY THE CFT77 COMPILER ON V1INP

This appendix contains a listing of the error messages that were returned when the original input processor source deck was compiled using the FORTRAN CFT77 compiler. The code was compiled using a FORTRAN compiler CFT when it was developed by Boeing. Error messages will lead to fatal crashes when decks containing them are tried to execute. Warning messages may lead to fatal crashes and cautions are usually just noteworthy comments.

Use this listing or others that can be generated for the library decks and the solver code setting ON=H in the CFT77 job control statement under COS. These can be compared to the successful executable decks to trace down any changes the author made to the EM-TRANAIR code.

*****APPENDIX B - OUTPUT GENERATED FROM CFT77 COMPILATION OF THE ORGINAL INPUT PROCESSOR SOURCE DECK V1INP*****

```
1          PAGE     1            Cray FORTRAN CFT77 3.1.2     02/14/90 18:58:38           05/23/90 08:32:02  PAGE     1
 (CPU=CRAY-XMP:EMA:CICS:VPOP,ON=AHPQR,OFF=CFGJOSMXZ,OPT=FULL:NOZEROINC,INTEGER=46,ADDRESS=FULL,ALLOC=STATIC)

      1      1.       PROGRAM AAINPUT                                                    AAINPUT.2
     75      1.       BLOCK DATA INIT                                                       INIT.2
    235      1.       SUBROUTINE FLDFIL(NK,K)                                             FLDFIL.2
    248      1.       SUBROUTINE ERRMSG(IFLAG,INAM,IERR)                                  ERRMSG.2
    437      1.       SUBROUTINE NETMOD(NNETT,NA,NL,L,K,I)                                NETMOD.2
    451      1.       SUBROUTINE OPDEF                                                     OPDEF.2
    466      1.       SUBROUTINE PIW4(IN,OUT,NW)                                            PIW4.2
    499      1.       SUBROUTINE UNPIW4 (IN,OUT,NW)                                       UNPIW4.2
 *** *** FF169 [warning ] < UNPIW4, Line = 9, File = $CPL, line = 507 > :
             Integer arithmetic operation exceeds 46-bit maximum.  Result will be computed with 64-bit arithmetic
    543      1.       SUBROUTINE BOPDEF(ITBX,IZMPT,ITPTL,NTPN,KTPN,NDPTS,NAPTS,           BOPDEF.2
 *** *** FF622 [warning ] < BOPDEF, Line = 190, File = $CPL, line = 732 > :
             OPTIMIZATION - an exponentiation was replaced with a multiplication.  This may cause numerical differences.
    777      1.       SUBROUTINE EDGPNT(P1,P2,P,P21,V,Q,D)                                EDGPNT.2
    799      1.       SUBROUTINE FBFND(P,PI,DP,DF,IMIN,IMAX,JMIN,JMAX,KMIN,KMAX)           FBFND.2
    824      1.       SUBROUTINE FESBLE(N,A,B,C,CHECK)                                    FESBLE.2
    853      1.       SUBROUTINE FLDDEF(NX,NY,NZ,PI,DP,MATF,S,KB,ERABRT)                  FLDDEF.2
    960      1.       SUBROUTINE LPCUT(Q1,Q2,CP,ICS,DCP,EN,ITR,R,P)                        LPCUT.2
   1023      1.       SUBROUTINE LTCUT(Q1,Q2,P1,P2,P3,R,P,CHECK)                           LTCUT.2
   1049      1.       SUBROUTINE PANDQC(Z1,Z2,Z3,Z4,CP,ICS,DCP,EN,AR,AJ,ALAM,NOP)         PANDQC.2
   1105      1.       SUBROUTINE PANDEF                                                   PANDEF.2
   1238      1.       SUBROUTINE PANPNT(CP,ICS,EN,P,ITR,ISED,Q,D)                         PANPNT.2
   1348      1.       SUBROUTINE PFBLST(PI,DP,DF,CP,ICS,NX,NY,NZ,NB,KB)                   PFBLST.2
   1390      1.       SUBROUTINE PIFBCH(XYZ,CP,ICS,CHECK)                                 PIFBCH.2
   1411      1.       SUBROUTINE TADJST(P1,P2,P3,PTOL1,P,Q,ADJST)                         TADJST.2
   1436      1.       SUBROUTINE TCOORD(P1,P2,P3,P,P21,P31,EN,Q)                          TCOORD.2
   1468      1.       SUBROUTINE TIFBCH(XYZ,P1,P2,P3,CHECK)                               TIFBCH.2
   1495      1.       SUBROUTINE TRIPNT(P1,P2,P3,P,PS,PT,PH,V,Q,D)                        TRIPNT.2
   1533      1.       SUBROUTINE STRTAU(NJP,TAU1,TAU2,TAUEMP,MSEG)                        STRTAU.2
   1612      1.       SUBROUTINE ABTCHK(K,NMK,NNK,NZAK,ISD,                               ABTCHK.2
   1690      1.       SUBROUTINE CPETP(NMK,NNK,NZAK,ZM,ISD,IZ1,IZ2,Z,IZM)                  CPETP.2
   1710      1.       SUBROUTINE EDGABT(NML,NNL,NZAL,JSD,JZ1,JZ2,                         EDGABT.2
   1784      1.       SUBROUTINE DETEQV(NPEABT, NOTOK)                                    DETEQV.2
   1973      1.       SUBROUTINE PROEQV(NPEABT, IECTOT)                                   PROEQV.2
   2145      1.       SUBROUTINE CHGGRD(NPEABT)                                           CHGGRD.2
   2362      1.       SUBROUTINE PRONEQ(NPEABT, IECTOT)                                   PRONEQ.2
   2568      1.       SUBROUTINE PEADET(NPEABT,IE,KZEDG,KNCEDG,KNCINT,KNEDG)              PEADET.2
   2687      1.       SUBROUTINE NETWRKS                                                 NETWRKS.2
```

```
2734    1.    SUBROUTINE  MSHIND(ISD,IZ,IN,NM,NN,L)                                   MSHIND.2
2791    1.    SUBROUTINE EMARK  (LABEL)                                                 EMARK.2
2800    1.    SUBROUTINE FBXLST(PI,DP,DF,IPAN,LPAN,KPAN,NPNT,IPNT,LPNT,KEY,NLST)       FBXLST.2
2874    1.    SUBROUTINE PADJST(K,L,CP,ICS,DCP,EN,PTOL1,PTOL2,PTOL3,P,Q)               PADJST.2
2995    1.    SUBROUTINE PANGHB(K,L,ISED,NAP,KAP,LAP)                                  PANGHB.2
3135    1.    SUBROUTINE EPTAU(NP,IFDSGN,TAU,TAUTMP,EPSTAU)                             EPTAU.2
3159    1.    SUBROUTINE ABTIDN(NNETT,NM,NN,Z,NTD,COMPRS,BETAMS,EPSGEO,                ABTIDN.2
3788    1.    SUBROUTINE ABTDNC  (NNETT,NM,NN,Z,Q)                                     ABTDNC.2
3805    1.    SUBROUTINE DNCHEK  (KNET,M,N,Z,Q,HEADER)                                 DNCHEK.2
3846    1.    SUBROUTINE MPNORM  ( Z1,  Z2,  Z3,  Z4,  EN)                             MPNORM.2
3858    1.    SUBROUTINE D2LINE  (Z1,Z2,   DIST)                                       D2LINE.2
3958    1.    SUBROUTINE DOMVEC  (A,V)                                                 DOMVEC.2
3975    1.    SUBROUTINE ABTAIP  (NNETT,NM,NN,Z,NTD,COMPRS,EPSGEO,NSYMM,NZA            ABTAIP.2
4060    1.    SUBROUTINE A3TDZO  (NNETT,NM,NN,Z,EPSGEO,NZA,ZSV)                        ABTDZO.2
4093    1.    SUBROUTINE DZBCHK  (K,M,N,EPS,Z,ZSV,HEADER)                              DZBCHK.2
4199    1.    SUBROUTINE ABTECD(NNETT,NM,NN,Z,NTD,COMPRS,BETAMS,EPSGEO,NSYMM,NZA       ABTECD.2
4444    1.    SUBROUTINE ABTFSD  (NNETT,NM,NN,Z,NTD,COMPRS,EPSGEO,NSYMM,NZA            ABTFSD.2
4764    1.    SUBROUTINE ABTDAB  (NNETT,NM,NN,Z,NTD,COMPRS,EPSGEO,NSYMM,NZA            ABTDAB.2
4997    1.    SUBROUTINE ABTSYM  (NNETT,NM,NN,Z,NTD,COMPRS,EPSGEO,NSYMM,NZA            ABTSYM.2
5098    1.    SUBROUTINE ABTDAI  (NNETT,NM,NN,Z,NTD,COMPRS,EPSGEO,NSYMM,NZA            ABTDAI.2
5237    1.    SUBROUTINE ABTDUE(NNETT,NM,NN,Z,NTD,COMPRS,BETAMS,EPSGEO,NSYMM,NZA       ABTDUE.2
5561    1.    SUBROUTINE ABTAIO  (NNETT,NM,NN,Z,NTD,COMPRS,EPSGEO,NSYMM,NZA            ABTAIO.2
5833    1.    SUBROUTINE ABTABO  (NNETT,NM,NN,Z,NTD,COMPRS,EPSGEO,NSYMM,NZA            ABTABO.2
6017    1.    SUBROUTINE FSGCMP  (KMP,NEDMPA,NFSGA,KFDSEG,NNETT                        FSGCMP.2
6089    1.    SUBROUTINE ADDIN2  (N,IA,   IAX)                                         ADDIN2.2
6114    1.    SUBROUTINE ABTEMP  (NE,KFDS,KSGN,   Z,NZA,NM,NN,NEDMPA,   EPSGEO         ABTEMP.2
6376    1.    SUBROUTINE ABTEQC  (KPT,KSGN,NPT,   IX,JX,IJSGN)                         ABTEQC.2
6441    1.    SUBROUTINE MPTEQC  (KPT,NPT,   IX,JX)                                    MPTEQC.2
6492    1.    SUBROUTINE ABTMSG  (MSG)                                                 ABTMSG.2
6504    1.    SUBROUTINE ABTEND  (MSG)                                                 ABTEND.2
6517    1.    SUBROUTINE CMPIED  (KMP,   NNETT,NEDMPA,NZA,NM,NN,   KZ)                 CMPIED.2
6553    1.    SUBROUTINE EDGIND  (ISD,NMK,NNK,   KZEDG,KNCEDG,KNCINT,KNEDG)            EDGIND.2
6587    1.    SUBROUTINE EPOINT  (Z,INCZ,NZ,   T,ZT)                                   EPOINT.2
6603    1.    SUBROUTINE NRMESH  (ZX,   ZE,INCE,NE,   ISGN,TE,   ZNR,INR,DNR)          NRMESH.2
6629    1.    SUBROUTINE ABTELS  (P1,P2,   ZE,INCE,NE,   EPSGEO,   ABUT,TE1,TE2)     ABTELS.2
6670    1.    SUBROUTINE IDNGEO  (Z1,Z2,EPSGEO,   EPSEQU)                              IDNGEO.2
6682    1.    SUBROUTINE ABTPOS  (Z,EPSGEO,NSYMM,   KPOS)                              ABTPOS.2
6691    1.    SUBROUTINE EDGMPI  (KEDG,IMP,NEDMPA,   IEDMP)                            EDGMPI.2
6723    1.    SUBROUTINE NREDGE  (P,  ZE,INCE,NE,  ISGN,TSGN,  PE,TE,DE)               NREDGE.2
6822    1.    SUBROUTINE INTNRM  (Z1,Z2,Y1,Y2,   EN)                                   INTNRM.2
6836    1.    SUBROUTINE EDGTAU  (ZE,INCE,NE,DZE,   TE,   TAUE)                        EDGTAU.2
6870    1.    SUBROUTINE OUTPKV  (LABEL,N,V)                                           OUTPKV.2
6891    1.    SUBROUTINE BMARK  (LABEL)                                                 BMARK.2
6900    1.    SUBROUTINE PERPNT(P1,P2,P3,P4,Z,Q,D,RMIN,DS)                             PERPNT.2
```

```
6965     1.        SUBROUTINE FILOPN                                                  FILOPN.2
6981     1.        SUBROUTINE KEYINV(M,K,N,L)                                         KEYINV.2
6989     1.        SUBROUTINE OPGEN                                                    OPGEN.2
7020     1.        SUBROUTINE PEAFOR( NPEABT, NOTOK )                                 PEAFOR.2
7387     1.        SUBROUTINE PROMIN(Z,Q,V1,V2,R)                                     PROMIN.2
7433     1.        SUBROUTINE LSOLVE(M,N,L,A,INFO)                                    LSOLVE.2
7446     1.        SUBROUTINE MINCUT(P1,P2,IPM,ITRM,IUPLM,PM)                         MINCUT.2
7488     1.        SUBROUTINE GRDGEN(OX,OY,OZ,XX,XY,XZ,YX,YY,YZ,ZX,ZY,ZZ,DX,DY,DZ,    GRDGEN.2
7569     1.        SUBROUTINE MSPNT1                                                  MSPNT1.2
7691     1.        SUBROUTINE CAMBER(KN,NPCT,NYST,NCEN,NTRL)                          CAMBER.2
7853     1.        SUBROUTINE CIRC(K)                                                   CIRC.2
7917     1.        SUBROUTINE ELLPT(K)                                                 ELLPT.2
7987     1.        SUBROUTINE GADNET(KN,NROW,NCOL,NCEN)                               GADNET.2
8069     1.        SUBROUTINE TRINP(IERR)                                              TRINP.2
9977     1.        SUBROUTINE INBC(K,ICA,BET,NBET)                                      INBC.2
10425    1.        SUBROUTINE INBC1(ICA,K,BET,NBET)                                    INBC1.2
10490    1.        SUBROUTINE INBC2(ICA,K,BET,NBET)                                    INBC2.2
10553   64.          READ(NIN,5070)  (RHS(1,L+I),I=1,NS)                               INBC2.41

*** *** FF133 [warning ] < INBC2, Line = 64, File = $CPL, line = 10553 > :
             The number of subscripts is less than the number of declared dimensions
 10560    71.              RHS(1,L+I)=BET(1,LM+I)                                     INBC2.48

*** *** FF133 [warning ] < INBC2, Line = 71, File = $CPL, line = 10560 > :
             The number of subscripts is less than the number of declared dimensions
 10562    73.            RHS(1,L+M)=RHS(1,L+NMM1)                                     INBC2.50

*** *** FF133 [warning ] < INBC2, Line = 73, File = $CPL, line = 10562 > :
             The number of subscripts is less than the number of declared dimensions

*** *** FF133 [warning ] < INBC2, Line = 73, File = $CPL, line = 10562 > :
             The number of subscripts is less than the number of declared dimensions
 10566    77.          CALL SCOPY(M,RHS(1,L-M+1),2,RHS(1,L+1),2)                      INBC2.54

*** *** FF133 [warning ] < INBC2, Line = 77, File = $CPL, line = 10566 > :
             The number of subscripts is less than the number of declared dimensions

*** *** FF133 [warning ] < INBC2, Line = 77, File = $CPL, line = 10566 > :
             The number of subscripts is less than the number of declared dimensions
 10568    79.        CALL RZERO(M*N,RHS(2,ICA+1),2)                                   INBC2.56

*** *** FF133 [warning ] < INBC2, Line = 79, File = $CPL, line = 10568 > :
             The number of subscripts is less than the number of declared dimensions
 10569    80.        IF(AMNSW.EQ.1.) CALL MTRXTP(2,M,N,RHS(1,ICA+1),BET,4*NBET)       INBC2.57
```

```
*** *** FF133 [warning ] < INBC2, Line = 80, File = $CPL, line = 10569 > :
                 The number of subscripts is less than the number of declared dimensions
  10584      1.       SUBROUTINE MESHP(K,IPTER,AMNSW,DNSMSH,S,NS)                          MESHP.2
  10794      1.       SUBROUTINE QUADNT(KN,NROW,NCOL)                                     QUADNT.2
  10878      1.       SUBROUTINE TRWAKE(KN,XWAKE)                                         TRWAKE.2
  10972      1.       SUBROUTINE PEAINP(NPEABT)                                           PEAINP.2
  11182      1.       SUBROUTINE MSHDNS(ZM,NM,NN,MF,NF,NOUT)                              MSHDNS.2
  11291      1.       SUBROUTINE INECHO(IERR)                                             INECHO.2
  11335      1.       SUBROUTINE INSUM                                                     INSUM.2
  11595      1.       SUBROUTINE IJKLAT(IJK,I,J,K)                                        IJKLAT.2
  11607      1.       SUBROUTINE LATIJK(I,J,K,IJK)                                        LATIJK.2
  11616      1.       SUBROUTINE GRDDEF                                                   GRDDEF.2
  11662      1.       SUBROUTINE ASEGL(KNET,KSD,IMP,NNSM,NSMLST,MXLST)                     ASEGL.2
  11746      1.       SUBROUTINE ABTSEG(NABT,NEDABA,NEDMP,KFDKEY,KFDSEG,TAUEMP,           ABTSEG.2
  11996      1.       SUBROUTINE APNTL(KNET,KSD,IMP,NNSM,NSMLST,MXLST)                     APNTL.2
  12082      1.       SUBROUTINE ABTPT1(NABT,NEDABA,NEDMP,KFDKEY,KFDSEG,TAUEMP,           ABTPT1.2
  12301      1.       SUBROUTINE ABTPT2(NABINT,NMPAIA,NEDMP,KEMKEY,KFDSEG,                ABTPT2.2
  12543      1.       SUBROUTINE CHCOLP(KFDSEG,KSG1,KSG2,MSEG,NCOLPS,KCOLPS)              CHCOLP.2
  12570      1.       SUBROUTINE BCONCL                                                   BCONCL.2
  12947    378.       CALL RFILL(4,RHS(1,K),BET,1)                                        BCONCL.287

*** *** FF133 [warning ] < BCONCL, Line = 378, File = $CPL, line = 12947 > :
                 The number of subscripts is less than the number of declared dimensions
  12954    385.       CALL RFILL(4,RHS(2,K),BET,1)                                        BCONCL.294

*** *** FF133 [warning ] < BCONCL, Line = 385, File = $CPL, line = 12954 > :
                 The number of subscripts is less than the number of declared dimensions
  13417      1.       SUBROUTINE SPBSC (KC,IFN,JFN,ISD,  KBASIC,  LOCSRT,KEYLOC,MAPS)      SPBSC.2
  13461      1.       SUBROUTINE BCOPT                                                     BCOPT.2
  13779      1.       SUBROUTINE BLCAL(BLCP,BL)                                            BLCAL.2
  13864      1.       SUBROUTINE BLCCAL(M,N,NM,NN,ZM,Z,B)                                 BLCCAL.2
  14267      1.       SUBROUTINE CMNGRD (KNET,MCP,NCP)                                    CMNGRD.2
  14304      1.       SUBROUTINE CONTRL (KN,NT,NM,NN,NC,NCA,NBCA,NMAPCA,NPA               CONTRL.2
  14797      1.       SUBROUTINE CPABT (IFN,JFN,KN, KABMTC,KFSG,TAUC,ZNC,NEDG,KSD,IDCPMC   CPABT.2
  15124      1.       SUBROUTINE CPNOR(IP,ZP,ENP)                                          CPNOR.2
  15286      1.       SUBROUTINE DASPL(KNET,NTK,NM,NN,NXA,NSSA,NS,NSS,NPA,ZM)              DASPL.2
  15778      1.       SUBROUTINE XDASPL (KNET,IPAN,JPAN,  INDX,IIDX,ASTX,  NSSAX,IPRNT)   XDASPL.2
  15993      1.       SUBROUTINE XDASIN (ASTD,IID,IND,  INDX,IIDX,ASTX,  IPRNT)           XDASIN.2
  16052      1.       SUBROUTINE EDGSPL (X,W,A)                                           EDGSPL.2
  16076      1.       SUBROUTINE GEOMC(KN,NM,NN,NPA,NZA,ZM,NASRAT)                         GEOMC.2
  16567      3.       SUBROUTINE REFLOC (EN,SFAC,  A,JAC,RPNTYP,  AI)                     REFLOC.2
  16648      1.       SUBROUTINE SINCD(Z,DS,IC)                                            SINCD.2
  16789      1.       SUBROUTINE SING (KNET,NTK,NM,NN,NSA,NSSA,NS,NSS                       SING.2
```

```
17191       1.        SUBROUTINE SNGDEL                                                      SNGDEL.2
17583       1.        SUBROUTINE SUBPQR(CP,AR,P,ALAM,PP,QQ,RR,IC)                            SUBPQR.2
17649       1.        SUBROUTINE SURPRO(Z,ZP,IC)                                             SURPRO.2
17771       1.        SUBROUTINE TCNTRL                                                      TCNTRL.2
17984       1.        SUBROUTINE TSING                                                        TSING.2
18194       1.        SUBROUTINE ZCADJ                                                        ZCADJ.2
18395       3.        SUBROUTINE ENRCHG(K,M,N,Z)                                             ENRCHG.2
18441       1.        SUBROUTINE GTALAM (C1,C2,ALAM)                                         GTALAM.2
18475       1.        SUBROUTINE NRPTHP (FTCMPD,CP,ICS,PZ,  SVAL,TVAL)                       NRPTHP.2
18682       1.        SUBROUTINE INSIDE (Q,ICS,ANM,P,WITHIN)                                 INSIDE.2
18717       1.        SUBROUTINE NRPTED (QM,QP,P,QNEAR,DIST,TAU)                             NRPTED.2
18750       1.        SUBROUTINE OPSTNC(NX,NY,NZ,IBVH,IBDW)                                  OPSTNC.2
18787       1.        SUBROUTINE DEFOPG(RDXYZ,DXYZ,BETAMS,OPVH,OPRT,OPDW,OPDFS,OPDWL)        DEFOPG.2
18854       1.        SUBROUTINE OPSET                                                        OPSET.2
18904      51.        EQUIVALENCE (KB,IPNT), (S,KPAN), (MATF,KPNT), (IFPTL,IPNT)              OPSET.21
*** *** FF726 [warning ] < OPSET > :
              Array MATF with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < OPSET > :
              Array S with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the
              array.
*** *** FF726 [warning ] < OPSET > :
              Array IFPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < OPSET > :
              Array KB with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the
              array.
*** *** FF430 [error   ] < OPSET > :
              Array IFPTL is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < OPSET > :
              Array KB is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < OPSET > :
              Array S is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < OPSET > :
              Array MATF is assumed-size but is not a dummy argument or a pointee.
19000       1.        SUBROUTINE DEFDPT                                                      DEFDPT.2
19055      56.        2          (IAD,IAPTL(6*MAXSRC+1)), (KAD,ITPN)                         DEFDPT.34
*** *** FF726 [warning ] < DEFDPT > :
              Array IAS with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array MATG with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
```

```
              Array IAPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array IAD with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array ITPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array IDT with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array BUFBC with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array DIVD with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array ISPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array ITPN with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array KAD with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array MDPT with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array IDPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array IZMPT with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array S with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to the
              array.
*** *** FF726 [warning ] < DEFDPT > :
              Array IDN with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
*** *** FF726 [warning ] < DEFDPT > :
              Array KDN with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
              the array.
```

```
*** *** FF726 [warning ] < DEFDPT > :
        Array KTPN with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
        the array.
*** *** FF726 [warning ] < DEFDPT > :
        Array MAP with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
        the array.
*** *** FF430 [error   ] < DEFDPT > :
        Array KAD is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array IAD is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array KDN is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array IDN is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array IZMPT is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array ISPTL is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array IAS is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array IDPTL is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array MDPT is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array CDPT is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array IAPTL is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array MAP is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array BUFBC is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array S is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array DIVD is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array IPND is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array KDT is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array IDT is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
        Array KTPN is assumed-size but is not a dummy argument or a pointee.
```

```
*** *** FF430 [error   ] < DEFDPT > :
            Array ITPN is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
            Array ITPTL is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < DEFDPT > :
            Array MATG is assumed-size but is not a dummy argument or a pointee.
 19234      1.       SUBROUTINE APTDEL(NTPTS,NDPTS,NAPTS,ITPTL,IDT,KDT,IDN,KDN,         APTDEL.2
 19316      1.       SUBROUTINE DEFFPT(MATG,NTPTS,ITPTL,NFPTS,IFPTL,KGRD)              DEFFPT.2
 19388      1.       SUBROUTINE TFILL(IOPT,N,IF1,NT,IFT,IFS)                            TFILL.2
 19404      1.       SUBROUTINE ADJWK(M,N,ZI,ZO,ISIDE,X2IN,XFPE,NADD)                   ADJWK.2
 19494      1.       SUBROUTINE DREGD(IPND,NDPTS,MDPT,KEY)                              DREGD.2
 19600      1.       SUBROUTINE VOPCAL(IAD,ID,IA,NA,NDN,KDN,                           VOPCAL.2
 19866      1.       SUBROUTINE MAPINV(N,MAP,M,IPNT,MAPI,KEY)                          MAPINV.2
 19891      1.       SUBROUTINE EQUIVC(N,MAP,NEQ,IEQ,KEQ,KEY)                          EQUIVC.2
 19919      1.       SUBROUTINE LSCHOP(K,P,DP,P1,P2,T1,T2)                             LSCHOP.2
 19945      1.       SUBROUTINE EIFBCH(P,DP,P1,P2,T1,T2)                               EIFBCH.2
 19988      1.       SUBROUTINE CIFBCH(P,DP,P1,CHECK)                                  CIFBCH.2
 19999      1.       SUBROUTINE TRICAL(PH,DH,P1,P2,P3,P,NE)                            TRICAL.2
 20054      1.       SUBROUTINE ECAL(P1,P2,EN,E,NEC,F)                                   ECAL.2
 20112      1.       SUBROUTINE FCAL(NP,P,EN,F,NFC,E)                                    FCAL.2
 20134      1.       SUBROUTINE HCAL(NP,P,EN,H,NHC,F)                                    HCAL.2
 20151      1.       SUBROUTINE SOPCAL(NDPTS,NDIVD,DIVD,KDT,NS,S)                      SOPCAL.2
 20296    146.       IRHS=RHS(2*L,M,JP)/2                                              SOPCAL.111

*** *** FF251 [caution ] < SOPCAL, Line = 146, File = $CPL, line = 20296 > :
            A floating point division has been encountered in an expression being converted to integer.
 20298    148.       IRHS=RHS(2*L,M,JP+NM(KP))/2                                       SOPCAL.113

*** *** FF251 [caution ] < SOPCAL, Line = 148, File = $CPL, line = 20298 > :
            A floating point division has been encountered in an expression being converted to integer.
 20300    150.       IRHS=RHS(2*L,M,JP+NM(KP)+1)/2                                     SOPCAL.115

*** *** FF251 [caution ] < SOPCAL, Line = 150, File = $CPL, line = 20300 > :
            A floating point division has been encountered in an expression being converted to integer.
 20302    152.       IRHS=RHS(2*L,M,JP+1)/2                                            SOPCAL.117

*** *** FF251 [caution ] < SOPCAL, Line = 152, File = $CPL, line = 20302 > :
            A floating point division has been encountered in an expression being converted to integer.
 20543      1.       SUBROUTINE PDRVR (S,MXS)                                           PDRVR.2
 20617      1.       SUBROUTINE GLOCAT (IND,IX,IY,IZ)                                  GLOCAT.2
 20629      1.       SUBROUTINE VELCAL (VC,IC,ID,IB,IBI,IBB,IFPM,                      VELCAL.2
 20719      1.       SUBROUTINE TBTOP(NPANT,IPANT,NSCR,ISCR)                            TBTOP.2
 20816      1.       SUBROUTINE PEQCL(NPI,IPL,NPTOT,IPANT,MAXJPL,JPL,                   PEQCL.2
 20940      1.       SUBROUTINE PINSID(CP,P,DP,INSIDE)                                 PINSID.2
```

```
20973     1.     SUBROUTINE PNGHBR(IPL,INSIDE,MXLST,NSMLST,MAXJPL,NPJ,JPL,IULCON)    PNGHBR.2
21089     1.     SUBROUTINE INNGBR(NZA,INET,NM,NN,IS,IXP,IYP,IB,JISID,NPJ,JPL)       INNGBR.2
21141     1.     SUBROUTINE EXNGBR(INET,IS,IMP,MXLST,NSMLST,NNSM,NPJ,JPL,NPFLAG)     EXNGBR.2
21262     1.     SUBROUTINE REQC(NPEQC,LPEQC,IPEQC,IPLIST,NSEQC,NPI,ISEQC)             REQC.2
21496     1.     SUBROUTINE TREG(NPEQC,IPEQC,LPEQC,NSEQC,IEQC,                         TREG.2
21591     1.     SUBROUTINE PTINT(IP,IT,ITRI,PTRI,P)                                  PTINT.2
21630     1.     SUBROUTINE ABTORD(IABT,NEDABA,NE,IE,IFP,KSEG,KFDKEY,KFDSEG,         ABTORD.2
21794     1.     SUBROUTINE VMOMC(H,NH,D,LUMP,HH,EE)                                  VMOMC.2
21804     1.     SUBROUTINE EMNCUT(P1,P2,IP1,IT1,IP2,IT2,IPM,ITRM,IUPLM,PM)          EMNCUT.2
21849     1.     SUBROUTINE CMPCTL(N,IL1,M,IEQL,NL2,IL2)                             CMPCTL.2
21879     1.     SUBROUTINE OPPLOT(NX,NY,NZ,IF1,NFPTS,IFPTL,NTHER,NOUT)              OPPLOT.2
21916     1.     SUBROUTINE DEFPPT (KDT,IFPTL,IDT,IAPTL,TGRD,ITPTL,SCR,              DEFPPT.2
22280     1.     SUBROUTINE SMOMC(NOPT,F,NF,D,LUMP,EN,HH,EE,HNE,HNNH,ENNE,HP,EP)      SMOMC.2
22332     1.     SUBROUTINE LINTRI(P1,P2,P3,C)                                       LINTRI.2
22359     1.     SUBROUTINE INDCHK(K,N,L,I,M)                                        INDCHK.2
22370     1.     SUBROUTINE TRICPI(I,CP,P0,CPT)                                      TRICPI.2
22412     1.     SUBROUTINE FPNPLT(P,DP,NP,CP,ICS,IDP,K,S,NOUT)                      FPNPLT.2
22513     1.     SUBROUTINE SELUL(ISE,IP,IT,IUL,EN,ULSE,P)                            SELUL.2
22550     1.     SUBROUTINE GNRML(KNET,ZM,NM,NN,L,ZN,NOUT)                            GNRML.2
22614     1.     SUBROUTINE FLDCHK(MATG,NX,NY,NZ,NTPTS,ITPTL,KGRD)                   FLDCHK.2
22661     1.     SUBROUTINE TRINRM(I,CP,P0,ENC,ENV)                                  TRINRM.2
22673     1.     SUBROUTINE PAREA(NP,P,EN,AREA)                                       PAREA.2
22695     1.     SUBROUTINE TESTP(C,NP,IAPTL,PHE)                                     TESTP.2
22720     1.     SUBROUTINE OPCHK(IOPER)                                              OPCHK.2
22752    33.     DATA TOL /1.E-9/                                                     OPCHK.19
*** *** FF726 [warning ] < OPCHK > :
           Array OPER with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
           the array.
*** *** FF726 [warning ] < OPCHK > :
           Array BUFR with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
           the array.
*** *** FF726 [warning ] < OPCHK > :
           Array IAPTL with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
           the array.
*** *** FF726 [warning ] < OPCHK > :
           Array RFLD with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
           the array.
*** *** FF726 [warning ] < OPCHK > :
           Array DRHS with length 1 is EQUIVALENCEd to a COMMON block - CFT77 assumes there are no out-of-bounds references to
           the array.
*** *** FF430 [error   ] < OPCHK > :
           Array DRHS is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < OPCHK > :
           Array BUFR is assumed-size but is not a dummy argument or a pointee.
```

```
*** *** FF430 [error   ] < OPCHK > :
            Array OPER is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < OPCHK > :
            Array IAPTL is assumed-size but is not a dummy argument or a pointee.
*** *** FF430 [error   ] < OPCHK > :
            Array RFLD is assumed-size but is not a dummy argument or a pointee.
 22831     1.       SUBROUTINE MATCHK(NABT,NEDABA,KFDKEY,KFDSEG,                     MATCHK.2
 22944     1.       SUBROUTINE FAOPRT(OPER,FAOP,NOUT)                                FAOPRT.2
 22964    21.      C                OPER(I2+2*IW)),IW=1,NP)                          FAOPRT.21

*** *** FF323 [error   ] < FAOPRT, Line = 21, File = $CPL, line = 22964 > :
            Syntax error - input symbol is ")" but expected "operator or , or end of statement".
 22969    26.      C                OPER(I2+2*IW)),IW=1,NR)                          FAOPRT.26

*** *** FF323 [error   ] < FAOPRT, Line = 26, File = $CPL, line = 22969 > :
            Syntax error - input symbol is ")" but expected "operator or , or end of statement".
 22973     1.       SUBROUTINE AGNRML(ZM,NM,NN,L,ZNA,VEC1,VEC2,IOK)                  AGNRML.2
 23055     1.       SUBROUTINE CDRVR (S,MXS)                                          CDRVR.2
 23115     1.       SUBROUTINE CNTD (IX,IY,IZ,XC)                                      CNTD.2
 23125     1.       SUBROUTINE VCALF (VC,IC,ID,IB,IBI,IBIB,IM,SM,IBB,IBD,             VCALF.2
 23428     1.       SUBROUTINE VCALA (VC,IC,ID,IB,IBI,IBIB,IM,SM,IBB,IBD,             VCALA.2
 23771     1.       SUBROUTINE IBSET (IBI,IB,IBB,IBD,IBKD,IFPM,NX,NY)                 IBSET.2
 23925     1.       SUBROUTINE DEFCPT (VC,ITPTL,TGRD,IPND,IDPTL,IDT,                 DEFCPT.2
 24128     1.       SUBROUTINE LINLS(N,P,O,Q,X,NOUT)                                  LINLS.2
 24158     1.       SUBROUTINE CDRVR1 (S,MXS)                                        CDRVR1.2
 24218     1.       SUBROUTINE IBSET1 (IBI,IB,IBB,IFPM,NX,NY)                        IBSET1.2
 24398     1.       SUBROUTINE DFCPT1 (VC,ITPTL,TGRD,IPND,IDPTL,IDT,                 DFCPT1.2
 24615     1.       SUBROUTINE VCALAA (VC,IC,ID,IB,IBI,IBIB,IM,SM,IBB,IBD,           VCALAA.2
 24791     1.       SUBROUTINE LUMPC(H,KDT,HS,NXYZ3)                                  LUMPC.2
 24796     1.       SUBROUTINE LUMPL(J1,J2,J3,J4,C)                                   LUMPL.2
 24804     1.       SUBROUTINE LUMPS(C)                                               LUMPS.2
 24838     1.       SUBROUTINE LSQSFT(ZK,WTK,AK,NO,NPK)                              LSQSFT.2
 24870     1.       SUBROUTINE BASEVL(P,D,H,E)                                       BASEVL.2
 24876     1.       SUBROUTINE BASISQ(NOPT,LUMP,F,NF,D,P,H,E,HH,EE,EH,HP,EP)         BASISQ.2
 25085     1.       SUBROUTINE ABTINT (ISYM,LABT                                     ABTINT.2
 25392     1.       SUBROUTINE GPHSCN (NB,NNODE,P,Q,BRNM,PR,KB,NETWK,  NTR,NBTRA     GPHSCN.2
 25563     1.       SUBROUTINE GPHPLK (NB,  NNODE,MNOD,  P,Q,BRNM,W,KB               GPHPLK.2
 25716     1.       SUBROUTINE GPLUCK (NB,NNODE,MNOD,IGRD                            GPLUCK.2
 25851     1.       SUBROUTINE DERIVE                                                DERIVE.2
 25862     1.       SUBROUTINE SITPNT(IP,IT,IN,P)                                    SITPNT.2
 25898     1.       SUBROUTINE THNBCN(NPOL,NMAT,ALPHA,BETA,MATL,MATU,NMATI,          THNBCN.2
 25986     1.       SUBROUTINE TKCAL (NPOL,FREQ,WN0,NMAT,EPSMU,ALPHA,BETA)            TKCAL.2
 26007     1.       SUBROUTINE TREBRN(MP,NP,N,MAP)                                   TREBRN.2
 26036     1.       SUBROUTINE TRERUT(N,MAP)                                         TRERUT.2
```

```
26048      1.        SUBROUTINE KID(NX,NY,NZ,NZMPT,NTPTS,ITPTL,NDPTS,IDPTL,IDT,          KID.2
26156      1.        SUBROUTINE FACRDC(NP,P,IDP,PTOL,NS,KS)                            FACRDC.2
26186      1.        SUBROUTINE FACTOP(NP,P,IDP,IFC,PTOL,NS,S)                         FACTOP.2
26199      1.        SUBROUTINE TDPEQV(NP,P,IFC,PTOL,NEQ,LEQ,IEQ,KEQ,KS)               TDPEQV.2
26234      1.        SUBROUTINE ODPEQV(NP,P,PTOL,NEQ,LEQ,IEQ,KEQ)                      ODPEQV.2
26250      1.        SUBROUTINE ANGORD(NP,P,IP,KS)                                     ANGORD.2
26293      1.        SUBROUTINE EDGRDC(NS,IS,JS,P,IDP,PTOL)                            EDGRDC.2
26325      1.        SUBROUTINE PLNEQV(NP,P,IDP,IFC,NEQ,LEQ,IEQ,KEQ,S)                 PLNEQV.2
26378      1.        SUBROUTINE PROJD(P0,P1,P2,D)                                       PROJD.2
26391      1.        SUBROUTINE SOUT                                                     SOUT.2
26552      1.        SUBROUTINE WRFAOP(NFA,NFDW,NBLEM,LBUFF,BUFF,NDFOUT)               WRFAOP.2
26571      1.        SUBROUTINE WRSTOP(NDFOUT,WN0S,NMAT,ALPHA,BETA,NELSTD,TOP,STDOPS)  WRSTOP.2
26600      1.        SUBROUTINE FREEOP(RDXYZ,WN0S,NELSTD,TOP)                          FREEOP.2
26639      1.        SUBROUTINE BDYST(IBIAS,NXYZ,MAXBDY,IBBDYE,IBBDYR,NELBDY)           BDYST.2
26725      1.        SUBROUTINE WVCAL(ANG,DX,DY,DZ,WVM,WV,EV,HV)                        WVCAL.2
26808      1.        SUBROUTINE CAMLT2(A,B,C,NOP)                                      CAMLT2.2
26827      1.        SUBROUTINE TZCAL(D,ALPHA,BETA,T,NOPT)                              TZCAL.2
26844      1.        SUBROUTINE CORIND(NX,NY,NZ,N,I,J,K,L,M)                           CORIND.2
26854      1.        SUBROUTINE FLDSCT(NF,IF,NS,IS)                                    FLDSCT.2
26861      1.        SUBROUTINE FSDCOF(WN0,DI,COF)                                     FSDCOF.2
26903      1.        SUBROUTINE HNGHBR(IG,JG,KG,MG,I,L)                                HNGHBR.2
26919      1.        SUBROUTINE MXMT(A,N1,B,N2,C,N3)                                     MXMT.2
26924      1.        SUBROUTINE OPCOMP(OPER,NCOF,COFP,IP,COFR,IR,KEY,SCALE)            OPCOMP.2
27002      1.        SUBROUTINE DMATOP(ALP0,BET0,ALP,BET,OPD0,OPD)                     DMATOP.2
27014      1.        SUBROUTINE EMOUT(NOP)                                              EMOUT.2
27124    111.       C                 ((MATI(I,N),DI(I,N)),I=1,NMATI(N))                EMOUT.75

*** *** FF498 [error   ] < EMOUT, Line = 111, File = $CPL, line = 27124 > :
          Component of the complex constant is not a real or an integer constant, or, this is an I/O statement which contains
          illegal parenthesis'.
27153      1.        SUBROUTINE OPADD(L,KDT,IAD,KAD,IAPTS,NXYZ3,DIVD)                   OPADD.2
27200      1.        SUBROUTINE LSTDEF(IOPER)                                          LSTDEF.2
27305      1.        SUBROUTINE CONDEF(NX,NY,NZ,MATF,NFAKI,                            CONDEF.2
27322      1.        SUBROUTINE CJDEF(NX,NY,NZ,MATF,NCON,ICONL,IRECCB,                  CJDEF.2
27363      1.        SUBROUTINE SRCDEF(NX,NY,NZ,MATF,NFAKI,NFPTL,MAXLST,               SRCDEF.2
27406      1.        SUBROUTINE RHODEF(NX,NY,NZ,MATF,MAXLST,IBIAS,                     RHODEF.2
27426      1.        SUBROUTINE GRHDEF(NX,NY,NZ,MATF,MAXLST,IBIAS,                     GRHDEF.2
27448      1.        SUBROUTINE GAMDEF(NX,NY,NZ,MATF,NCON,ICONL,IRECCB,                GAMDEF.2
```

APPENDIX C

ORIGINAL AND REVISED INITIALIZATION FOR V1INP

The two decks listed in this appendix were used to initialize variables for the input processor V1INP source program AAINPUT. The first listing is from the original code where a call statement in AAINPUT is used to call up the BLOCK DATA INIT. The second listing is the result of solving an execution error by imbedding the initialization in the source program AAINPUT.

*****APPENDIX C - INITIALIZATION OF PARAMETERS FROM ORIGINAL INPUT PROCESSOR DECK V1INP**

```
*DECK INIT
      BLOCK DATA INIT
*CA PARAM
*CA /CHRCTR/
*CALL PRNT
*CALL ABTPRT
*CALL ABTNEW
*CA /CRSDAT/
*CA /FILDAT/
*CA /GRIDQ/
*CA /INCFEM/
*CA /ITERQ/
*CALL /ARVIS/
*CA /NUMRCN/
*CA /PHYSCN/
*CA /LSCALE/
*CALL ACASE
*CALL COMPRS
*CALL SYMM
*CALL /OFBOD/
*CALL KSTMLN
*CALL FMCOF
C
C     INITIALIZE FUNDAMENTAL COMMON BLOCK VARIABLES
C
C     /CHRCTR/
      DATA REVDAT/'7/31/87'/
      DATA TITLE /'  DEFAULT TITLE
     1                                '/,  NAME /'  DEFAULT NAME
     2                                                 '/,
     3     IPW /'PW  '/, IPOL /'EFLD'/
C     /PRNT/
      DATA IGEOMP /0/, ISINGP /0/, ICONTP /0/, IBCONP /0/
      DATA IEDGEP /0/, ISINGS /0/, IPRAIC /0/, IPARTP /0/
      DATA IPARTS /0/, NEXDGN /0/, IOUTPR /0/, IFMCPR /0/
      DATA ICOSTP /0/, IEXTRP /0/, ISPMAP /0/, ICPMAP /0/
      DATA IBCMAP /0/
C     /ABTPRT/
      DATA IGEOIN /1/, IGEOUT /0/, NWXREF /0/, NWPROP /0/, IABUTD /0/
      DATA IABSUM /1/, IFABPR /1/
      DATA IGEOM /0/, IBCND /0/, IOPER /0/, IPANL /0/
      DATA NWDK /0/
C     /ABTNEW/
      DATA EPSGEO /0./
C     /ACASE/
      DATA ALPHA /4*0./, BETA /4*0./, FSVM /4*1./
      DATA IACASE /1/, NACASE /1/
C     /COMPRS/
      DATA AMACH /0.0001/, ALPC /0./, BETC /0./
C     /SYMM/
      DATA NSYMM /0/, MISYM /0/, MJSYM /0/
C     /OFBOD/
      DATA NOF /0/
```

```
C     /KSTMLN/
      DATA NUMPTS /0/
C     /FMCOF/
      DATA SREF /1./, BREF /1./, CREF /1./, DREF /1./
      DATA XREF /0./, YREF /0./, ZREF /0./
C     /FILDAT/
      DATA NIN /5/, NOUT /6/, NPRSL /7/, NPRPO /8/
      DATA NABFL /11/, NFAKI /12/, NFBLP /13/, NFBKP /14/,
     .     NFVD/15/,NFVC/16/,NFBC/17/,NFGRE/18/,NFGRD/19/,
     .     NFDWF/20/,NFDWA/21/,NFDWD/22/,NFPTL/23/,NFMTL/24/,
     .     NFSCR/25/
C     /CRSDAT/
      DATA NABSW /1/, NANGB(1) /1/, ANGB(1,1) /0./, ANGB(2,1) /0./,
     2     ANGB(3,1) /0./
C     /GRIDQ/
      DATA NX /NGRD1/, NY /NGRD2/, NZ /NGRD3/
      DATA XI /0./, XF /0./, YI /0./, YF /0./, ZI /0./, ZF /0./
C     /INCFEM/
      DATA FREQ/3.E8/,EHMAG/(1.,0.)/,
     2     NASW /1/, NANG(1) /1/, ANG(1,1) /0./, ANG(2,1) /0./,
     3     ANG(3,1) /0./
C     /ITERQ/
      DATA TOL /1.E-10/, NITER /50/, NSRCH /50/, MSGLVL /0/
      DATA RLOSS /0./, DROPT /0./, NRELXD /1/
      DATA IDMRCH /3,5*0/, IDSWP /2,5*0/
C     /ARVIS/
      DATA NVIS /1/, RMCUT /.95/
C     /LSCALE/
      DATA RLEN /1./
C     /NUMRCN/
      DATA PI /3.141592653358979/, CONE /(0.,1.)/

C     /PHYSCN/
      DATA GAMMA /1.4/
      DATA VLIGHT/2.997925E8/
      END
```

*****APPENDIX C - INITIALIZATION IMBEDDED IN INPUT PROCESSOR V1INP SOURCE PROGRAM AAINPUT***

```
*DECK AAINPUT
      PROGRAM AAINPUT
C
C     DRIVER FOR THE TRANAIR INPUT PROCESSOR
C
*CA /CASE/
*CA /FILDAT/
*CALL ABTPRT
*CALL /CHRCTR/
*CALL /GRIDQ/
*CA PARAM
*CALL PRNT
*CALL ABTNEW
*CA /CRSDAT/
*CA /INCFEM/
*CA /ITERQ/
*CALL /ARVIS/
*CA /NUMRCN/
*CA /PHYSCN/
*CA /LSCALE/
*CALL ACASE
*CALL COMPRS
*CALL SYMM
*CALL /OFBOD/
*CALL KSTMLN
*CALL FMCOF
      REAL CARD(10)
C
C     INITIALIZE FUNDAMENTAL COMMON BLOCK VARIABLES
C
C     /CHRCTR/
      DATA REVDAT/'7/31/87'/
      DATA TITLE /'  DEFAULT TITLE
     1                          '/,  NAME /'  DEFAULT NAME
     2                                                    '/,
     3     IPW /'PW  '/, IPOL /'EFLD'/
C     /PRNT/
      DATA IGEOMP /0/, ISINGP /0/, ICONTP /0/, IBCONP /0/
      DATA IEDGEP /0/, ISINGS /0/, IPRAIC /0/, IPARTP /0/
      DATA IPARTS /0/, NEXDGN /0/, IOUTPR /0/, IFMCPR /0/
      DATA ICOSTP /0/, IEXTRP /0/, ISPMAP /0/, ICPMAP /0/
      DATA IBCMAP /0/
C     /ABTPRT/
      DATA IGEOIN /1/, IGEOUT /0/, NWXREF /0/, NWPROP /0/, IABUTD /0/
      DATA IABSUM /1/, IFABPR /1/
      DATA IGEOM /0/, IBCND /0/, IOPER /0/, IPANL /0/
      DATA NWDK /0/
C     /ABTNEW/
      DATA EPSGEO /0./
C     /ACASE/
      DATA ALPHA /4*0./, BETA /4*0./, FSVM /4*1./
      DATA IACASE /1/, NACASE /1/
C     /COMPRS/
```

```
      DATA AMACH /0.0001/, ALPC /0./, BETC /0./
C     /SYMM/
      DATA NSYMM /0/, MISYM /0/, MJSYM /0/
C     /OFBOD/
      DATA NOF /0/
C     /KSTMLN/
      DATA NUMPTS /0/
C     /FMCOF/
      DATA SREF /1./, BREF /1./, CREF /1./, DREF /1./
      DATA XREF /0./, YREF /0./, ZREF /0./
C     /FILDAT/
      DATA NIN /5/, NOUT /6/, NPRSL /7/, NPRPO /8/
      DATA NABFL /11/, NFAKI /12/, NFBLP /13/, NFBKP /14/,
     .     NFVD/15/,NFVC/16/,NFBC/17/,NFGRE/18/,NFGRD/19/,
     .     NFDWF/20/,NFDWA/21/,NFDWD/22/,NFPTL/23/,NFMTL/24/,
     .     NFSCR/25/
C     /CRSDAT/
      DATA NABSW /1/, NANGB(1) /1/, ANGB(1,1) /0./, ANGB(2,1) /0./,
     2     ANGB(3,1) /0./
C     /GRIDQ/
      DATA NX /NGRD1/, NY /NGRD2/, NZ /NGRD3/
      DATA XI /0./, XF /0./, YI /0./, YF /0./, ZI /0./, ZF /0./
C     /INCFEM/
      DATA FREQ/3.E8/,EHMAG/(1.,0.)/,
     2     NASW /1/, NANG(1) /1/, ANG(1,1) /0./, ANG(2,1) /0./,
     3     ANG(3,1) /0./
C     /ITERQ/
      DATA TOL /1.E-10/, NITER /50/, NSRCH /50/, MSGLVL /0/
      DATA RLOSS /0./, DROPT /0./, NRELXD /1/
      DATA IDMRCH /3,5*0/, IDSWP /2,5*0/
C     /ARVIS/
      DATA NVIS /1/, RMCUT /.95/
C     /LSCALE/
      DATA RLEN /1./
C     /NUMRCN/
      DATA PI /3.141592653358979/, CONE /(0.,1.)/
C     /PHYSCN/
      DATA GAMMA /1.4/
      DATA VLIGHT/2.997925E8/
C
      WRITE(NOUT,6005)
C
      CALL DERIVE
C
      CALL VHEAD(NOUT,'V1INP:',REVDAT)
      WRITE(NOUT,6030)
C
C
      CALL CSTPRT('INITIAL')
C
      IERR=0
      CALL TRINP(IERR)
C
      CALL CSTPRT('TRINP')
      IF(IERR.EQ.3) THEN
      WRITE(NOUT,6060)
```

```
      CALL ABORT('END-OF-FILE ON INPUT')
      ENDIF
C
      IF(IERR.NE.0) THEN
      WRITE(NOUT,6050) IERR
      CALL ABORT
      ENDIF
      CALL OPDEF
      CALL CSTPRT('OPDEF')
      DO 130 I=1,4
        IF(IDCHK(I).EQ.-1) GO TO 200
  130 CONTINUE
C
      CALL SOUT
      CALL CSTPRT('SOUT')
C
  200 CONTINUE
C
      READ(NIN,END=210) CARD
      GOTO 200
C
  210 CONTINUE
C
      REWIND NPRSL
      REWIND NPRPO
C
C
      WRITE(NOUT,6040)
C
C
 6005 FORMAT(///10X,'ENTERING INITIALIZATION PHASE')
 6030 FORMAT(///10X,' STARTING INPUT PROCESSOR PROGRAM')
 6040 FORMAT(//10X,'COMPLETING INPUT PROCESSOR PROGRAM')
 6050 FORMAT(//10X,'PROGRAM ABORT DUE TO ERROR',I4)
 6060 FORMAT(//10X,'END-OF-FILE ON INPUT FILE')
      STOP
      END
```

APPENDIX D

ORIGINAL AND REVISED CONDUCTING SPHERE NETWORK INPUT DATA

This appendix contains the original and revised listings for the conducting sphere configuration. The original network configuration used by Boeing contains twice as many points as the executable network.

*****APPENDIX D - ORIGINAL CONDUCTING SPHERE SURFACE GRID NETWORK INPUT DATA***

```
$TITLE
SPHERE - NO PLANES OF SYMMETRY
CASE #1
$SYM
0.        0.
$EAT
0.0        -1.      0.0       0.0       0.0
$DAT
0.        +1.       0.        0.        0.        0.0
$BOX
 -1.0000   1.0000    17.
 -1.0000   1.0000    17.
 -1.0000   1.0000    17.
$NET
 4.
$CIR
 4.
 1.                                     1.        0.0
 0.                                                     RIGHT FRONT
41.
.8        0.        .79938323 .03140785 .79753387 .06276728
.79445477 .09402992 .79015067 .12514757 .78462822 .15607226
.77789594 .18675629 .76996419 .21715236 .76084521 .24721360
.75055307 .27689365 .73910363 .30614675 .72651454 .33492779
.71280522 .36319240 .69799681 .39089699 .68211213 .41799885
.66517569 .44445619 .64721360 .47022820 .62825355 .49527516
.60832477 .51955844 .58745801 .54304060 .56568543 .56568543
.54304060 .58745801 .51955844 .60832477 .49527516 .62825355
.47022820 .64721360 .44445619 .66517569 .41799885 .68211213
.39089699 .69799681 .36319240 .71280522 .33492779 .72651454
.30614675 .73910363 .27689365 .75055307 .24721360 .76084521
.21715236 .76996419 .18675629 .77789594 .15607226 .78462822
.12514757 .79015067 .09402992 .79445477 .06276728 .79753387
.03140785 .79938323 0.0       .8
81.
90.       92.25     94.5      96.75     99.0      101.25
103.5     105.75    108.      110.25    112.5     114.75
117.      119.25    121.5     123.75    126.      128.25
130.5     132.75    135.0     137.25    139.5     141.75
144.      146.25    148.5     150.75    153.      155.25
157.5     159.75    162.0     164.25    166.5     168.75
171.      173.25    175.5     177.75    180.      182.25
184.5     186.75    189.      191.25    193.5     195.75
198.      200.25    202.5     204.75    207.0     209.25
211.5     213.75    216.0     218.25    220.5     222.75
225.      227.25    229.5     231.75    234.0     236.25
238.5     240.75    243.0     245.25    247.5     249.75
252.0     254.25    256.5     258.75    261.0     263.25
265.5     267.75    270.0
0.                                                     LEFT FRONT
41.
0.        .8        -.03140785.79938323 -.06276728.79753387
-.09402992.79445477 -.12514757.79015067 -.15607226.78462822
```

```
-.18675629.77789594 -.21715236.76996419 -.24721360.76084521
-.27689365.75055307 -.30614675.73910363 -.33492779.72651454
-.36319240.71280522 -.39089699.69799681 -.41799885.68211213
-.44445619.66517569 -.47022820.64721360 -.49527516.62825355
-.51955844.60832477 -.54304060.58745801 -.56568543.56568543
-.58745801.54304060 -.60832477.51955844 -.62825355.49527516
-.64721360.47022820 -.66517569.44445619 -.68211213.41799885
-.69799681.39089699 -.71280522.36319240 -.72651454.33492779
-.73910363.30614675 -.75055307.27689365 -.76084521.24721360
-.76996419.21715236 -.77789594.18675629 -.78462822.15607226
-.79015067.12514757 -.79445477.09402992 -.79753387.06276728
-.79938323.03140785 -.8        0.0
81.
90.       92.25     94.5      96.75     99.0      101.25
103.5     105.75    108.      110.25    112.5     114.75
117.      119.25    121.5     123.75    126.      128.25
130.5     132.75    135.0     137.25    139.5     141.75
144.      146.25    148.5     150.75    153.      155.25
157.5     159.75    162.0     164.25    166.5     168.75
171.      173.25    175.5     177.75    180.      182.25
184.5     186.75    189.      191.25    193.5     195.75
198.      200.25    202.5     204.75    207.0     209.25
211.5     213.75    216.0     218.25    220.5     222.75
225.      227.25    229.5     231.75    234.0     236.25
238.5     240.75    243.0     245.25    247.5     249.75
252.      254.25    256.5     258.75    261.0     263.25
265.5     267.75    270.0
 0.                                                         RIGHT BACK
41.
.8        0.        .79938323 .03140785 .79753387 .06276728
.79445477 .09402992 .79015067 .12514757 .78462822 .15607226
.77789594 .18675629 .76996419 .21715236 .76084521 .24721360
.75055307 .27689365 .73910363 .30614675 .72651454 .33492779
.71280522 .36319240 .69799681 .39089699 .68211213 .41799885
.66517569 .44445619 .64721360 .47022820 .62825355 .49527516
.60832477 .51955844 .58745801 .54304060 .56568543  56568543
.54304060 .58745801 .51955844 .60832477 .49527516 .62825355
.47022820 .64721360 .44445619 .66517569 .41799885 .68211213
.39089699 .69799681 .36319240 .71280522 .33492779 .72651454
.30614675 .73910363 .27689365 .75055307 .24721360 .76084521
.21715236 .76996419 .18675629 .77789594 .15607226 .78462822
.12514757 .79015067 .09402992 .79445477 .06276728 .79753387
.03140785 .79938323 0.0       .8
81.
270.      272.25    274.5     276.75    279.      281.25
283.5     285.75    288.      290.25    292.5     294.75
297.      299.25    301.5     303.75    306.0     308.25
310.5     312.75    315.      317.25    319.5     321.75
324.      326.25    328.5     330.75    333.      335.25
337.5     339.75    342.      344.25    346.5     348.75
351.      353.25    355.5     357.75    000.0     002.25
004.5     006.75    009.0     011.25    013.5     015.75
018.0     020.25    022.5     024.75    027.0     029.25
031.5     033.75    036.0     038.25    040.5     042.75
045.0     047.25    049.5     051.75    054.0     056.25
058.5     060.75    063.0     065.25    067.5     069.75
```

```
072.0     074.25    076.5     078.75    81.       83.25
85.5      87.75     90.
0.                                                      LEFT BACK
41.
0.        .8        -.03140785.79938323 -.06276728.79753387
-.09402992.79445477 -.12514757.79015067 -.15607226.78462822
-.18675629.77789594 -.21715236.76996419 -.24721360.76084521
-.27689365.75055307 -.30614675.73910363 -.33492779.72651454
-.36319240.71280522 -.39089699.69799681 -.41799885.68211213
-.44445619.66517569 -.47022820.64721360 -.49527516.62825355
-.51955844.60832477 -.54304060.58745801 -.56568543.56568543
-.58745801.54304060 -.60832477.51955844 -.62825355.49527516
-.64721360.47022820 -.66517569.44445619 -.68211213.41799885
-.69799681.39089699 -.71280522.36319240 -.72651454.33492779
-.73910363.30614675 -.75055307.27689365 -.76084521.24721360
-.76996419.21715236 -.77789594.18675629 -.78462822.15607226
-.79015067.12514757 -.79445477.09402992 -.79753387.06276728
-.79938323.03140785 -.8       0.0
81.
270.      272.25    274.5     276.75    279.      281.25
283.5     285.75    288.      290.25    292.5     294.75
297.      299.25    301.5     303.75    306.0     308.25
310.5     312.75    315.      317.25    319.5     321.75
324.      326.25    328.5     330.75    333.      335.25
337.5     339.75    342.      344.25    346.5     348.75
351.      353.25    355.5     357.75    000.0     002.25
004.5     006.75    009.0     011.25    013.5     015.75
018.0     020.25    022.5     024.75    027.0     029.25
031.5     033.75    036.0     038.25    040.5     042.75
045.0     047.25    049.5     051.75    054.0     056.25
058.5     060.75    063.0     065.25    067.5     069.75
072.0     074.25    076.5     078.75    81.0      83.25
85.5      87.75     90.0
$ITE
50.       5.        0.        0.        50.       1.
0.
0.        0.
$INC
HFLD
PW
1.7893    8.        1.
1.
1.
0.        0.        0.        0.        0.        0.
$RCS
2.
91.
0.        0.        0.        2.        0.        0.
91.
0.        -90.      90.       2.        0.        0.
$THE
4.
1.        1.        2.        0.
2.        1.        2.        0.
3.        1.        2.        0.
4.        1.        2.        0.
```

```
$PRI
1.        0.        0.        0.
1.        0.        0.        0.
$END
/EOF
```

*****APPENDIX D - EXECUTABLE CONDUCTING SPHERE NETWORK WITH FEWER SURFACE GRID POINTS***

```
$TITLE
SPHERE - NO PLANES OF SYMMETRY
CASE #1
$SYM
0.        0.
$EAT
0.0       0.0       0.0       0.0       0.0
$DAT
0.        0.0       0.        0.        0.        0.0
$BOX
 -1.0000   1.0000    17.
 -1.0000   1.0000    17.
 -1.0000   1.0000    17.
$NET
 4.
$CIR
 4.
 1.                                     1.        0.0
 0.                                                          RIGHT FRONT
41.
.8        0.        .79938323 .03140785 .79753387 .06276728
.79445477 .09402992 .79015067 .12514757 .78462822 .15607226
.77789594 .18675629 .76996419 .21715236 .76084521 .24721360
.75055307 .27689365 .73910363 .30614675 .72651454 .33492779
.71280522 .36319240 .69799681 .39089699 .68211213 .41799885
.66517569 .44445619 .64721360 .47022820 .62825355 .49527516
.60832477 .51955844 .58745801 .54304060 .56568543 .56568543
.54304060 .58745801 .51955844 .60832477 .49527516 .62825355
.47022820 .64721360 .44445619 .66517569 .41799885 .68211213
.39089699 .69799681 .36319240 .71280522 .33492779 .72651454
.30614675 .73910363 .27689365 .75055307 .24721360 .76084521
.21715236 .76996419 .18675629 .77789594 .15607226 .78462822
.12514757 .79015067 .09402992 .79445477 .06276728 .79753387
.03140785 .79938323 0.0       .8
41.
90.       94.5      99.0      103.5     108.      112.5
117.      121.5     126.      130.5     135.0     139.5
144.      148.5     153.      157.5     162.0     166.5
171.      175.5     180.      184.5     189.      193.5
198.      202.5     207.0     211.5     216.0     220.5
225.      229.5     234.0     238.5     243.0     247.5
252.0     256.5     261.0     265.5     270.0
0.                                                           LEFT FRONT
41.
0.        .8        -.03140785.79938323 -.06276728.79753387
-.09402992.79445477 -.12514757.79015067 -.15607226.78462822
-.18675629.77789594 -.21715236.76996419 -.24721360.76084521
-.27689365.75055307 -.30614675.73910363 -.33492779.72651454
-.36319240.71280522 -.39089699.69799681 -.41799885.68211213
-.44445619.66517569 -.47022820.64721360 -.49527516.62825355
-.51955844.60832477 -.54304060.58745801 -.56568543.56568543
-.58745801.54304060 -.60832477.51955844 -.62825355.49527516
-.64721360.47022820 -.66517569.44445619 -.68211213.41799885
```

```
-.69799681.39089699 -.71280522.36319240 -.72651454.33492779
-.73910363.30614675 -.75055307.27689365 -.76084521.24721360
-.76996419.21715236 -.77789594.18675629 -.78462822.15607226
-.79015067.12514757 -.79445477.09402992 -.79753387.06276728
-.79938323.03140785 -.8       0.0
41.
90.       94.5      99.0      103.5     108.      112.5
117.      121.5     126.      130.5     135.0     139.5
144.      148.5     153.      157.5     162.0     166.5
171.      175.5     180.      184.5     189.      193.5
198.      202.5     207.0     211.5     216.0     220.5
225.      229.5     234.0     238.5     243.0     247.5
252.0     256.5     261.0     265.5     270.0
 0.                                                      RIGHT BACK
41.
.8        0.        .79938323 .03140785 .79753387 .06276728
.79445477 .09402992 .79015067 .12514757 .78462822 .15607226
.77789594 .18675629 .76996419 .21715236 .76084521 .24721360
.75055307 .27689365 .73910363 .30614675 .72651454 .33492779
.71280522 .36319240 .69799681 .39089699 .68211213 .41799885
.66517569 .44445619 .64721360 .47022820 .62825355 .49527516
.60832477 .51955844 .58745801 .54304060 .56568543 .56568543
.54304060 .58745801 .51955844 .60832477 .49527516 .62825355
.47022820 .64721360 .44445619 .66517569 .41799885 .68211213
.39089699 .69799681 .36319240 .71280522 .33492779 .72651454
.30614675 .73910363 .27689365 .75055307 .24721360 .76084521
.21715236 .76996419 .18675629 .77789594 .15607226 .78462822
.12514757 .79015067 .09402992 .79445477 .06276728 .79753387
.03140785 .79938323 0.0       .8
41.
270.      274.5     279.      283.5     288.      292.5
297.      301.5     306.0     310.5     315.      319.5
324.      328.5     333.      337.5     342.      346.5
351.      355.5     000.0     004.5     009.0     013.5
018.0     022.5     027.0     031.5     036.0     040.5
045.0     049.5     054.0     058.5     063.0     067.5
072.0     076.5     81.       85.5      90.
0.                                                       LEFT BACK
41.
0.        .8        -.03140785.79938323 -.06276728.79753387
-.09402992.79445477 -.12514757.79015067 -.15607226.78462822
-.18675629.77789594 -.21715236.76996419 -.24721360.76084521
-.27689365.75055307 -.30614675.73910363 -.33492779.72651454
-.36319240.71280522 -.39089699.69799681 -.41799885.68211213
-.44445619.66517569 -.47022820.64721360 -.49527516.62825355
-.51955844.60832477 -.54304060.58745801 -.56568543.56568543
-.58745801.54304060 -.60832477.51955844 -.62825355.49527516
-.64721360.47022820 -.66517569.44445619 -.68211213.41799885
-.69799681.39089699 -.71280522.36319240 -.72651454.33492779
-.73910363.30614675 -.75055307.27689365 -.76084521.24721360
-.76996419.21715236 -.77789594.18675629 -.78462822.15607226
-.79015067.12514757 -.79445477.09402992 -.79753387.06276728
-.79938323.03140785 -.8       0.0
41.
270.      274.5     279.      283.5     288.      292.5
297.      301.5     306.0     310.5     315.      319.5
```

```
324.      328.5     333.      337.5     342.      346.5
351.      355.5     000.0     004.5     009.0     013.5
018.0     022.5     027.0     031.5     036.0     040.5
045.0     049.5     054.0     058.5     063.0     067.5
072.0     076.5     81.       85.5      90.
$ITE
50.       5.        0.        0.        50.       1.
0.
0.        0.
$INC
HFLD
PW
1.7893    8.        1.
1.
1.
0.        0.        0.        0.        0.        0.
$RCS
4.
180.
0.        0.        0.        2.        0.        0.
180.
0.        -90.      90.       2.        0.        0.
-90.
0.        0.        0.        2.        0.        0.
-90.
0.        -90.      90.       2.        0.        0.
$THE
4.
1.        1.        2.        0.
2.        1.        2.        0.
3.        1.        2.        0.
4.        1.        2.        0.
$PRI
1.        1.        1.        1.
1.        1.        1.        1.
$END
/EOF
```

APPENDIX E

ORIGINAL AND REVISED INITIALIZATION FOR V1SOL

The two decks listed in this appendix were used to initialize variables for the solver V1SOL source program AA3DS. The first listing is from the original code where a call statement in AA3DS calls subroutine INIT which in turn contains a call statement to call up the BLOCK DATA INICOM. The second listing is the result of solving an execution error by imbedding the initialization in the source program AA3DS.

*****APPENDIX E - INITIALIZATION OF PARAMETERS WITH SUBROUTINES FROM THE ORIGINAL SOLVER DECK V1SOL*************************************

```
*DECK INICOM
      BLOCK DATA INICOM
C
C     BLOCK DATA INITIALIZATION
C
*CALL FORMARK
C
*CALL PARAM
C
*CALL COMMARK
C
*CALL /CBPARM/
*CALL /CHRDAT/
*CALL /DIAGN/
*CALL /FILDAT/
*CALL /GRIDQ/
*CALL /INCANG/
*CALL /INCFLQ/
*CALL /ITERQ/
*CALL /MATDAT/
*CALL /NUMRCN/
*CALL /PHYSCN/
C
C     INITIALIZE FUNDAMENTAL COMMON BLOCK VARIABLES
C
C     /CBPARM/
C
C     /CHRDAT/
      DATA REVDAT/'7/31/87'/
      DATA TITLE /'*DEFAULT *DEFAULT *DEFAULT *DEFAULT'/
      DATA NAME  /'*DEFAULT *DEFAULT *DEFAULT *DEFAULT'/
      DATA CHPOL/'EE'/,CHPW/'PW'/
C
C     /DIAGN/
      DATA EPS/1.E-8/,LIMIT/0/,NDOUT/6/
      DATA DEBUG/.FALSE./
C
C     /FILDAT/
      DATA NIN/5/,NOUT/6/,NERR/6/,NPRSL/7/,NFCJ/9/
      DATA NFRCS/10/
      DATA NDFOUT/4/
      DATA NFGRN/11/,IFSRCH/12/,IFOP/13/
      DATA IFNJ/14/,IFJ/15/,IFGRH/16/,IFGAM/17/
      DATA IFW/18/,IFDJ/19/
      DATA NFMRCS/29/
      DATA IFQC/20/,IFQB/21/,IFQCI/22/,IFQBI/23/,IFQS/24/
      DATA NFQW/25/,NFQWI/26/,IFQV/27/
      DATA IFQE /30/, IFQSC /31/, IFQSB /32/
      DATA NFIEL/28/
      DATA ISEQ/1/,IRAN/2/,IMEM/3/
C
C     /GRIDQ/
      DATA NX /17/, NY /17/, NZ /17/
```

```
      DATA XI /0./, XF /1./, YI /0./, YF /1./, ZI /0./, ZF /1./
C
C     /INCANG/
      DATA NASW/1/,NANG(1)/1/
C
C     /INCFLQ/
      DATA AANG(1) /0./, AANG(2) /0./, AANG(3) /0./
      DATA FREQ /3.E+8/, EMAG /(1.,0.)/
C
C     /ITERQ/
      DATA TOL /1.E-10/
      DATA NITER /30/, ITPRT /0/, KITER /20/, MSGLVL/3/
C
C     /MATDAT/
      DATA NMAT/2/
      DATA ALPHA(1)/(0.,0.)/,ALPHA(2)/(0.,0.)/
      DATA BETA(1)/(0.,0.)/,BETA(2)/(0.,0.)/
C
C     /NUMRCN/
      DATA CONE /(0.,1.)/
C
C     /PHYSCN/
      DATA VLIGHT /2.997925E8/
C
      END


*DECK INIT
      SUBROUTINE INIT
C
C     INITIALIZE DATA
C
*CALL FORMARK
C
*CALL COMMARK
C
*CALL LOCMARK
C
*CALL CODMARK
C
      CALL INICOM
C
      RETURN
      END
```

*****APPENDIX E - INITIALIZATION OF PARAMETERS IMBEDDED IN SOLVER V1SOL SOURCE PROGRAM AA3DS**

```
*DECK AA3DS
      PROGRAM AA3DS
C
C
C
*CALL FORMARK
C
*CALL COMMARK
C
*CALL PARAM
C
*CALL LOCMARK
C
*CALL CODMARK
C
*CALL /CBPARM/
*CALL /CHRDAT/
*CALL /DIAGN/
*CALL /FILDAT/
*CALL /GRIDQ/
*CALL /INCANG/
*CALL /INCFLQ/
*CALL /ITERQ/
*CALL /MATDAT/
*CALL /NUMRCN/
*CALL /PHYSCN/
*CALL /SCRTCH/
*CALL /STDOP/
*CALL /STENCL/
C
C
C
      CALL CSTPRT('BEGIN')
C
C
C     INITIALIZE DATA
C
C     BLOCK DATA INICOM
C
C     BLOCK DATA INITIALIZATION
C
C     INITIALIZE FUNDAMENTAL COMMON BLOCK VARIABLES
C
C     /CBPARM/
C
C     /CHRDAT/
      DATA REVDAT/'7/31/87'/
      DATA TITLE /'*DEFAULT *DEFAULT *DEFAULT *DEFAULT'/
      DATA NAME  /'*DEFAULT *DEFAULT *DEFAULT *DEFAULT'/
      DATA CHPOL/'EE'/,CHPW/'PW'/
C
C     /DIAGN/
      DATA EPS/1.E-8/,LIMIT/0/,NDOUT/6/
```

```
      DATA DEBUG/.FALSE./
C
C     /FILDAT/
      DATA NIN/5/,NOUT/6/,NERR/6/,NPRSL/7/,NFCJ/9/
      DATA NFRCS/10/
      DATA NDFOUT/4/
      DATA NFGRN/11/,IFSRCH/12/,IFOP/13/
      DATA IFNJ/14/,IFJ/15/,IFGRH/16/,IFGAM/17/
      DATA IFW/18/,IFDJ/19/
      DATA NFMRCS/29/
      DATA IFQC/20/,IFQB/21/,IFQCI/22/,IFQBI/23/,IFQS/24/
      DATA NFQW/25/,NFQWI/26/,IFQV/27/
      DATA IFQE /30/, IFQSC /31/, IFQSB /32/
      DATA NFIEL/28/
      DATA ISEQ/1/,IRAN/2/,IMEM/3/
C
C     /GRIDQ/
      DATA NX /17/, NY /17/, NZ /17/
      DATA XI /0./, XF /1./, YI /0./, YF /1./, ZI /0./, ZF /1./
C
C     /INCANG/
      DATA NASW/1/,NANG(1)/1/
C
C     /INCFLQ/
      DATA AANG(1) /0./, AANG(2) /0./, AANG(3) /0./
      DATA FREQ /3.E+8/, EMAG /(1.,0.)/
C
C     /ITERQ/
      DATA TOL /1.E-10/
      DATA NITER /30/, ITPRT /0/, KITER /20/, MSGLVL/3/
C
C     /MATDAT/
      DATA NMAT/2/
      DATA ALPHA(1)/(0.,0.)/,ALPHA(2)/(0.,0.)/
      DATA BETA(1)/(0.,0.)/,BETA(2)/(0.,0.)/
C
C     /NUMRCN/
      DATA CONE /(0.,1.)/
C
C     /PHYSCN/
      DATA VLIGHT /2.997925E8/
C
C     ***END OF INITIALIZATION DATA***
C
      CALL DATE(ADATE)
      CALL CLOCK(ATIME)
      CALL VHEAD(NOUT,'V1SOL:',REVDAT)
      CALL CSTPRT('INIT')
C
C     SET UP MEMORY MANAGER
      CALL SETRA(MAXSCR,SCRGLO)
C
      WRITE(NOUT,2)
    2 FORMAT(5X,'***JUST BEFORE CALL SUBROUTINE READIN***')
C
      CALL READIN(IREAD)
```

```
C
      WRITE(NOUT,3)IREAD
    3 FORMAT(5X,'***IREAD==',I5)
C
      WRITE(NOUT,4)
    4 FORMAT(5X,'***BACK FROM SUBROUTINE READIN***')
C
      CALL CSTPRT('READIN')
C
      IF (IREAD.NE.0) THEN
C
      WRITE(NOUT,6)
    6 FORMAT(5X,'***JUST BEFORE CALL SUBROUTINE SETUP***')
C
         CALL SETUP
C
      WRITE(NOUT,8)
    8 FORMAT(5X,'***BACK FROM SUBROUTINE SETUP***')
C
         CALL CSTPRT('SETUP')
C
      WRITE(NOUT,10)
   10 FORMAT(5X,'***JUST BEFORE CALL SUBROUTINE SOLVER***')
C
         CALL SOLVER
C
      WRITE(NOUT,12)
   12 FORMAT(5X,'***BACK FROM SUBROUTINE SOLVER***')
C
         CALL CSTPRT('SOLVER')
C
      WRITE(NOUT,14)
   14 FORMAT(5X,'***JUST BEFORE CALL SUBROUTINE OUTPUT***')
C
         CALL OUTPUT
C
      WRITE(NOUT,16)
   16 FORMAT(5X,'***BACK FROM SUBROUTINE OUTPUT***')
C
         CALL CSTPRT('OUTPUT')
C
      ENDIF
C
      STOP
      END
```

APPENDIX F

IEQUAL DUMMY SUBROUTINE

This appendix has the listing for subroutines CLUSS, CLUSM and IEQUAL which are all contained in V1SOL source program AA3DS. IEQUAL is called in both CLUSS and CLUSM to keep the memory space available for vector array CB and KCB intact.

*****APPENDIX F - SUBROUTINES CLUSS, CLUSM AND IEQUAL FROM SOLVER V1SOL USED TO PRESERVE VECTOR ARRAYS IN MEMORY*********************************

```
*DECK CLUSS
      SUBROUTINE CLUSS(N,NCB,KCB,CB,NQC,KC,NQB,KB,R,X)
C
*CA FORMARK
C
      DIMENSION R(2,N),X(2,N)
      DIMENSION KCB(NCB),CB(NCB),KC(*),KB(*)
C
C
*CA COMMARK
C
C
*CA LOCMARK
C
      COMPLEX Y
      DIMENSION Z(2)
C
C
*CA CODMARK
C
      CALL CCOPY(N,R,1,X,1)
      KCB(1)=0
      CALL FCHRB(NQC,KCB,NCB,KC,1,LC,MC,1)
      DO 50 J=1,N
      NREC=KCB(1)
      IREC=J-KCB(NCB)+1
      IF(IREC.LE.NREC) THEN
      MC=KCB(NCB-IREC)
      ELSE
      CALL FCHRB(NQC,CB,NCB,KC,J,LC,MC,1)
      ENDIF
      CALL IEQUAL(CB,KCB)
      NC=KCB(MC)
      JC=MC+NC+3
      Y=CMPLX(X(1,J),X(2,J))*CMPLX(CB(JC),CB(JC+1))
      X(1,J)=REAL(Y)
      X(2,J)=AIMAG(Y)
      IF(NC.GT.1) THEN
      Z(1)=X(1,J)
      Z(2)=X(2,J)
C     Z(1)=-Z(1)
C     Z(2)=-Z(2)
C     CALL CAXPYI(NC-1,Z,CB(JC+2),CB(MC+4),X)
C
      IK=MC+4
      M=JC+2
CDIR$ IVDEP
      DO 25 L=1,NC-1
      K=KCB(IK)
      X(1,K)=X(1,K)-Z(1)*CB(M)+Z(2)*CB(M+1)
      X(2,K)=X(2,K)-Z(1)*CB(M+1)-Z(2)*CB(M)
      M=M+2
      IK=IK+1
```

```
 25   CONTINUE
C
      ENDIF
 50   CONTINUE
      KCB(1)=0
      CALL FCHRB(NQB,KCB,NCB,KB,N,LB,MB,1)
      DO 90 I=1,N
      J=N+1-I
      IREC=J-KCB(NCB)+1
      IF(IREC.GE.1) THEN
      MB=KCB(NCB-IREC)
      ELSE
      CALL FCHRB(NQB,CB,NCB,KB,J,LB,MB,1)
      ENDIF
      CALL IEQUAL(CB,KCB)
      NB=KCB(MB)
      JB=MB+NB+3
      IB=MB+3*NB+1
      Y=CMPLX(X(1,J),X(2,J))*CMPLX(CB(IB),CB(IB+1))
      X(1,J)=REAL(Y)
      X(2,J)=AIMAG(Y)
      IF(NB.GT.1) THEN
      Z(1)=X(1,J)
      Z(2)=X(2,J)
C     Z(1)=-Z(1)
C     Z(2)=-Z(2)
C     CALL CAXPYI(NB-1,Z,CB(JB),CB(MB+3),X)
C
      IK=MB+3
      M=JB
CDIR$ IVDEP
      DO 75 L=1,NB-1
      K=KCB(IK)
      X(1,K)=X(1,K)-Z(1)*CB(M)+Z(2)*CB(M+1)
      X(2,K)=X(2,K)-Z(1)*CB(M+1)-Z(2)*CB(M)
      M=M+2
      IK=IK+1
 75   CONTINUE
C
      ENDIF
 90   CONTINUE
      RETURN
      END
*DECK CLUSM
      SUBROUTINE CLUSM(N,NCB,KCB,CB,NQC,KC,X,R)
C
*CA FORMARK
C
      DIMENSION R(2,N),X(2,N)
      DIMENSION KCB(NCB),CB(NCB),KC(*)
C
C
*CA COMMARK
C
C
*CA LOCMARK
```

```
C
      DIMENSION Z(2)
C
C
*CA CODMARK
C
      CALL CZERO(N,R,1)
      KCB(1)=0
      CALL FCHRB(NQC,KCB,NCB,KC,1,LC,MC,1)
      DO 50 J=1,N
      NREC=KCB(1)
      IREC=J-KCB(NCB)+1
      IF(IREC.LE.NREC) THEN
      MC=KCB(NCB-IREC)
      ELSE
      CALL FCHRB(NQC,CB,NCB,KC,J,LC,MC,1)
      ENDIF
      CALL IEQUAL(CB,KCB)
      NC=KCB(MC)
      JC=MC+NC+3
      Z(1)=X(1,J)
      Z(2)=X(2,J)
C     CALL CAXPYI(NC,Z,CB(JC),CB(MC+3),R)
C
      M=JC
      IK=MC+3
CDIR$ IVDEP
      DO 25 L=1,NC
      K=KCB(IK)
      R(1,K)=R(1,K)+Z(1)*CB(M)-Z(2)*CB(M+1)
      R(2,K)=R(2,K)+Z(1)*CB(M+1)+Z(2)*CB(M)
      M=M+2
      IK=IK+1
 25   CONTINUE
C
 50   CONTINUE
      RETURN
      END
*DECK IEQUAL
      SUBROUTINE IEQUAL(CB,KCB)
      RETURN
      END
```